Themis SPARC 5/64 Users Manual

Version 2.0 — January 15, 1996

Themis Computer—Americas and Pacific Rim
3185 Laurelview Court
Fremont, CA 94538
Phone (510) 252-0870
Fax (510) 490-5529
web site: *www.themis.com*

Themis Computer—Rest of World
1, Rue Des Essarts
Z.A. De Mayencin
38610 Gieres, France
Phone 33 76 59 60 61
Fax 33 76 63 00 30

©Copyright 1995 Themis Computer Inc.

All rights reserved. No part of this work may be reproduced in any form or by any means without prior written permission by Themis Computer.

Restricted Rights Legend: Use, duplication or disclosure by the US government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the *Rights in Technical Data and Computer Software* clause of *DFARS 252.227-7013* (October 1988) and *FAR 52.227-19* (June 1987).

The information in this document has been carefully checked and is believed to be accurate. However, Themis Computer assumes no responsibility for inaccuracies. Themis computer retains the right to make changes to this document at any time without prior notice. Themis Computer does not assume any liability arising from the application or use of this document or the products(s) described herein.

UNIX™ is a registered trademark of AT&T

OPENLOOK™ is a registered trademark of AT&T and Sun Microsystems

SOLARIS™ is a registered trademark of Sun Microsystems

SPARC™, SPARC 5/64™ are registered trademarks of SPARC International

The following are tardemarks of their respective companies and organizations. Themis Computer disclaims any responsibility for for specifying which marks are owned by which companies and organizations:

Advanced Micro Devices (AMD), ANSI, Centronics, IEEE, Intel, ISO, Ethernet, Fujitsu, JEDEC, Level One Communcations, Mbus, NewBridge, NICE, SBus, SCSI, Signetics, SGS Thompson Microelectronics, Sun Microsystems, VMEbus, Zilog

Table of Contents

1: How to Use This Manual *1-1*
1.1: Intended Audience 1-2
1.2: Organization 1-3
1.3: How To Start Quickly 1-5
1.4: In Case Of Difficulties 1-5
2: Introduction to the Themis SPARC 5/64 ..... *2-7*
2.1: Themis SPARC 5/64 Features 2-8
2.2: Themis SPARC 5/64 Supported Standards 2-10
3: Themis SPARC 5/64 Description *3-11*
3.1: Themis SPARC 5/64 Main Components 3-14
3.1.1: microSPARC II Microprocessor 3-14
Integer Unit (IU) and Floating-Point Unit (FPU) 3-14
Main Memory Cache 3-14
Reference Memory Management Unit (MMU) 3-15
Translation Lookaside Buffer (TLB) 3-15
RAM Controller 3-15
SBus Controller 3-15
3.1.2: Input/Output Controllers and 8-Bit Devices 3-17
MACIO (Master I/O Controller) 3-17
Ethernet Controller Macrocell 3-17
SCSI-2 Controller Macrocell 3-17
DMA Controller Macrocell 3-17
Printer Port Controller Macrocell 3-17
SLAVIO (Slave I/O Controller) 3-18
Serial Port A Controller 3-18
Serial Port B Controller 3-18
Floppy Disk Controller 3-18
Keyboard/Mouse Controller 3-18
EBus Controller 3-19
JTAG Internal & Boundary Scan Controller 3-19
Interrupt Controls 3-19

System Reset Controls 3-20
Counter/Timer Controls 3-20
SCV64 VMEbus Controller 3-20
KSIC SBus to SCV64 Interface 3-20
Slot 0: Extension SBus Slot 3-20
Single Inline Memory Modules (DRAM SIMMs) 3-21
Open Boot PROM (OBP) 3-23
Non-volatile RAM (NVRAM) 3-23
Time-of-Day Clock (TOD) 3-24
FLASH EPROM 3-24
3.1.3: 5/64 Board Connectors 3-24
SBus Slot (SBus Devices) Connectors 3-24
Main Memory DRAM SIMM Female Connectors 3-24
SCSI Female Connector 3-24
10base-T Ethernet 3-25
AUI Ethernet 3-25
Mouse/Keyboard Connector 3-25
Serial Port A&B 3-25
3.2: System Considerations 3-25
3.2.1: Required Initial Power-Up System Components 3-25
4: Themis SPARC 5/64 Specifications 4-27
4.1: Themis SPARC 5/64 Board 4-27
4.2: Themis SPARC 5/64 Electrical Specifications 4-31
4.3: Themis SPARC 5/64 Environmental Specifications 4-32
5: Themis SPARC 5/64 Installation 5-33
5.1: Registering Your SPARC 5/64 5-33
5.2: Unpacking The SPARC 5/64 5-34
5.3: Configuring the SPARC 5/64 5-35
5.4: OpenBOOT PROM Configuration 5-38
5.5: Configuring The VME Interface 5-41
5.5.1: VMEbus Memory Allocation 5-43
SBus Address Mapping 5-44
SBus Slot #0 5-44
SBus Slot #1 5-44
SBus Slot #2 5-44
SBus Slot #3 5-44
SBus slot #4 5-45
VME MMU Address Translation Table 5-45
5.6: Installing The 5/64 Paddle Board 5-47
5.7: Installing An SBus Card 5-47
5.8: Installing SIMM Memory 5-49
5.9: Attaching the 5/64 To A Network 5-52
5.10: Attaching a Keyboard and Mouse 5-53

6: Themis SPARC 5/64 Programmers Guide *6-55*
6.1: 5/64 VME Interface 6-56
6.1.1: Primary Features 6-56
6.1.2: System Controller 6-57
6.1.3: VMEbus Master Interface 6-58
Addressability 6-61
6.1.4: VME MMU Address Translation Table 6-61
6.1.5: VMEbus Slave Interface 6-65
6.1.6: VMEbus/SCV64 Interrupts 6-69
VME Interrupt Status 6-71
VMEbus Interrupt Acknowledgment 6-71
6.1.7: SCV64 Inter-Processor Communication/Location Monitor 6-72
6.2: Overview of VME Interface Under Solaris 1.1.1B/2.X 6-73
6.2.1: VMEbus Nexus Driver 6-74
6.2.2: Utility Drivers 6-75
6.2.3: Sample Drivers 6-76
6.2.4: OpenBoot PROM Echancements 6-77
A: Themis SPARC 5.64 Board Drawings *A-79*
KSIC Top A-79
KSIC Bottom A-80
SPARC 5/64 Top A-81
SPARC 5/64 Bottom A-82
B: Themis SPARC 5/64 Front Panels *B-83*
C: VME Access Speed *C-85*
D: VMEBus Interface Implementation *D-87*
D.1: Features D-88
D.2: General Descriptions D-90
D.3: SCV64 Registers D-91
D.4: VME System Controller D-92
D.4.1: Bus Arbitration D-92
Arbitration Modes D-92
Arbitration Time-out D-92
D.4.2: Bus Timer D-93
D.5: VME Slave Interface D-94
VMEBAR Register D-94
D.6: VME Interrupter D-95
VINT Register D-95
IVECT Register D-96
D.7: VME Interrupt Handling D-96
D.8: Location Monitor D-97
LMFIFO Location Monitor Read Port D-98
D.9: DMA Transfers D-98

DMALAR DMA Local Address Register D-98
DMAVAR DMA VMEbus Address Register D-98
DMATC DMA Transfer Count Register D-99
DMAVTC DMA VMEbus Transfer Count Register D-99
E: 5/64 Connector Specifications E-101
E.1: VME Bus Connectors (P1) E-102
E.2: VME Bus Printer Interface Connectors (P2) E-102
E.3: VME Bus Floppy Interface Connectors E-104
E.4: SBus Connectors Pinout J602 & J601 ..... E-105
E.5: Serial Ports A and B 26-pin Connector E-106
E.6: Keyboard/Mouse 8-pin DIN Connector E-106
E.7: SCSI Connector High Density 50-pin E-107
E.8: 10 Base-T RJ45 Twisted Pair Connector E-107
E.9: AUI Ethernet Connector E-108
F: Address Mapping Specifications F-109
F.1: Master VME Address Generation Overview F-109
F.2: Master VME MMU and SBus Slot Address Mapping F-110
F.3: microSPARC II Mapping F-112
F.4: DRAM Addressing Window F-113
F.5: SLAVIO EBus Mapping F-114
F.6: MACIO Mapping F-115
F.7: Rotary Switch Register F-116
F.8: User LED Register F-116
F.9: FLASH Programming Voltage Control Register F-116
F.10: FLASH Select Register F-117
F.11: KSIC/SCV64 Interrupt Status Register F-118
F.12: VME Interrupt Acknowledge Registers F-119
F.13: Local SCV64 Interrupt Acknowledge Registers F-120
F.14: VMMU Enable Register F-121
G: 5/64 Jumper Settings G-123
G.1: Factory Default Settings G-123
G.2: 5/64 Jumper Locations and Configuration G-125
H: Themis SPARC 5/64 Paddle Board H-135
I: OpenBoot PROM Extension Commands I-141
I.1: OBP Environment Variables I-141
I.2: OBP VME and Support Commands I-142
I.2.1: 5/64 OBP Command Description I-143
I.2.2: 5/64 OBP Environment Variables Description I-150

List of Figures

Figure 3-1 Layout of the Themis SPARC 5/64 Board 3-12
Figure 3-2 Block Diagram of the Themis SPARC 5/64 Board 3-13
Figure 3-3 SPARC 5/64 board - SIMM diagram 3-22
Figure 5-1 Attaching a Wrist Strap 5-34
Figure 5-2 5/64 default jumper settings (solder side) 5-36
Figure 5-3 5/64 default jumper setting (component side) 5-37
Figure 5-4 KSIC Interface default settings 5-38
Figure 5-5 BOOT PROM Mappings 5-40
Figure 5-6 microSPARC II Address Space Mapping Example. 5-43
Figure 5-7 SBus Slot #4 Resource Map 5-45
Figure 6-1 VME Master Address Generation 6-59
Figure 6-2 Default VME MMU Table 6-62
Figure 6-3 VME MMU Map Specifier Entry 6-63
Figure 6-4 VME MMU 16MB Segment Factory Defaults 6-64
Figure 6-5 VME Slave Address to Sbus Mapping 6-67
Figure 6-6 Slave Address Decoding 6-68
Figure 6-7 VME Interrupt Status Register 6-71

List of Tables

Table 3-1	Themis SPARC 5/64 SBus Slots	3-16
Table 3-2	microSPARC II General Mapping	3-16
Table 3-3	Interrupt Level Assignments.	3-19
Table 3-4	Approved list of vendors	3-22
Table 3-5	DC voltage as specified in the VME Specs.	3-25
Table 5-1	Approved list of vendors	5-50
Table 5-2	Memory Configuration	5-51
Table 6-1	Addressing Mode	6-59
Table 6-2	Data Transfer Mode	6-59
Table 6-3	Data Transfer Combinations	6-60
Table 6-4	Master Block Transfers	6-60
Table 6-5	DMA-Sourced VME Transfers	6-60
Table 6-6	Slave Window Size Configuration	6-66
Table 6-7	Solaris 1.1 Slave Accesses	6-68
Table 6-8	Solaris 2.x Slave Accesses	6-69
Table 6-9	SCV64 to SBus Interrupt Levels	6-70
Table 6-10	VMEbus Interrupt Mapping	6-75
Table 6-11	VME Device Drivers	6-75
Table D-1	Following is a list of the SCV64 Registers:	D-91
Table D-2	Arbitration Mode Settings	D-92
Table D-5	VINT Register Settings	D-95

How to Use This Manual

Thank you for purchasing the Themis 5/64 single board computer with VMEbus interface. Themis is a leading manufacturer of SPARC based processor board for the VMEbus market and designs and builds high quality products backed by responsive customer support.

This manual describes the Themis 5/64, a SPARCstation 5 compatible computer with VME64 VMEbus interface.

1.1 *Intended Audience*

It is written for end-users, integrators and programmers and contains all relevant information to install and configure the 5/64 board. It primarily assumes that the SUN Open Boot PROM (OBP) is installed. If you intend to operate the 5/64 with an operating system other than Solaris or use a real-time kernel such as VxWorks, please consult the appropriate documentation supplements accompanying your OS or kernel software.

Although all Themis 5/64 specific hardware and software features are described in detail in this manual, programmers wishing to code for the 5/64 without the benefit of an operating system or real-time kernel will require additional data sheets and documentation for the NCR chip-set, the microSPARC II processor and other system components comprising the SPARCstation 5 design.

The reader is assumed to be familiar with and have working knowledge of the VMEbus Rev. D specifications, SPARC processor architecture, NCR SPARCclassic chip-set, SBus A.2, Ethernet and SCSI (ANSI X3.131-1986).

For easy reference, all pertinent board information such as jumper layout and configuration, address maps, VMEbus accessing, etc. is available in the appendices.

1.2 *Organization*

This document is organized in chapters of increasing informational content and detail. A comprehensive set of appendices categorizes, concentrates and summarizes all 5/64 technical information necessary to efficiently configure and program the board. This document is divided into six chapters and nine appendices:

- **How to Use This Manual**, Chapter 1, contains a brief overview of this document and its organization.
- **Introduction to the Themis SPARC 5/64**, Chapter 2, describes the features and building blocks of the Themis 5/64. This chapter is intended as a brief overview of the 5/64 product and enumerates all major components of the board. It should be consulted to quickly gain an overview of the 5/64, its features and how it compares to its compatible SUN workstation products.
- **Themis SPARC 5/64 Description**, Chapter 3, is detailed description of the 5/64. Each major 5/64 component is described in detail. This chapter should be considered a primary source of component reference.
- **Themis SPARC 5/64 Specifications**, Chapter 4, lists all pertinent technical and environmental specification in tabular format. It is primarily intended for technical service personnel.
- **Themis SPARC 5/64 Installation**, Chapter 5, specifies instructions on how to install and configure the 5/64 for your particular environment and application. The contents of this chapter contains information mandatory for the correct operation of the 5/64 and should be consulted prior to its usage.
- **Themis SPARC 5/64 Programmers Guide**, Chapter 6, is for software engineers intend on programming the 5/64 at a low level and includes vital information on Themis specific features of the 5/64, such as the VME interface, OBP commands, etc.
- Appendix A, **Themis SPARC 5/64 Board Drawings**, contains board level drawing of the 5/64 single board computer.
- Appendix B, **Themis SPARC 5/64 Front Panels**, shows the front panel drawings of the single and double width versions of the 5/64 board.
- Appendix C, **VME Access Speed**, indicates the real measured VME data transfer speeds as obtained by Themis.

- Appendix D, **VMEBus Interface Implementation**, encompasses all VMEbus related information, including SCV64, custom VMEbus related registers, KSIC interface and VME MMU mapping.
- Appendix E, **5/64 Connector Specifications**, is a complete list of drawings, graphs and pin-outs on all 5/64 external connectors.
- Appendix F, **Addressing Mapping Specifications**, is the primary reference section on all pertinent device and register addresses. It shows all important addresses most likely used by programmers and engineers. VMEbus address generation and operation of the VME MMU translation table are also covered.
- Appendix G, **5/64 Jumper Settings**, is a comprehensive list of drawings and configuration options of all user modifiable jumpers and solder beads.
- Appendix H, **Themis SPARC 5/64 Paddle Board**, shows a drawing of the P2 paddle board as well as a listing and pin-out on all paddle board connectors.
- Appendix I, **OpenBoot PROM Extension Commands**, describes in detail all custom OpenBoot PROM commands implemented by Themis and primarily used in configuring the 5/64 computer.

1.3 How To Start Quickly

If you intend to start quickly with the 5/64 SPARC processor board, we recommend that you read the following information:

- Read Chapter 5, *Themis SPARC 5/64 Installation on page 5-33* in its entirety. It contains vital information on configuring the 5/64 and design and setup of complete VMEbus based systems.
- Consult Appendix I, *OpenBoot PROM Extension Commands on page I-141* for technical details on the Themis 5/64 specific configuration commands necessary to properly configure the processor board in a VMEbus environment.
- Appendix G, *5/64 Jumper Settings on page G-123* contains a complete list of all user configurable jumpers.

1.4 In Case Of Difficulties

If the 5/64 does not behave as described or if you encounter difficulties installing or configuring the 5/64 either standalone or as part of your VMEbus system environment, please call Themis Computer technical support at 510-252-0870, fax your questions to 510-490-5529, or e-mail them to *support@themis.com*. You can also contact us at our web site: *www.themis.com*.

Introduction to the Themis SPARC 5/64

The Themis SPARC 5/64 board is a SPARC [1]-based single-board VME computer, based on the new Sun/Fujitsu microSPARC® (aka SWIFT or microSPARC II), a highly integrated microprocessor with a wide array of on-chip features.

The 5/64 comes standard with two SBus slots, SCSI-2, Ethernet, Centronics, mouse/keyboard interface, 2 serial ports, VMEbus interface and JTAG testability for the major ASICs. The hardware chip set is composed of the microSPARC II, two NCR ASICs (SLAVIO and MACIO) and a VME64 interface using the Newbridge SCV64 controller and a Themis custom KSIC interface. The advanced VME interface supports fast 64-bit addressing and 64-bit data transfers and includes high speed DMA capabilities.

Use of the same chip set found in Sun's new SPARCstation 5 product line and standard Solaris 2.x CD-ROM binary distribution media assures compatibility with Sun Microsystems products on both hardware and software levels.

The Themis 5/64 SPARCstation 5 compatible CPU is a high performance, low cost VMEbus CPU board supporting Solaris 1.1.2 and Solaris 2.4 UNIX operating systems. For demanding real-time applications, a wide variety of real-time kernels including VxWorks and LynxOS are supported.

---

1. Scalable Processor Architecture Reduced-instruction-set Computer

2

The highly integrated Fujitsu / Sun microSPARC II processor was designed as a building block for cost effective computer systems with very low component counts. Themis has designed a powerful, simple and reliable CPU engine around the microSPARC II.

2.1 *Themis SPARC 5/64 Features*

Performance

- CPU: 112:5 Dhrystone MIPS/64 SPECin_92/54.6 SPECfp_92/ @ 85MHz
- FPU: 14.9 Linpack 1000 (with Apogee Compiler) MFLOPS @ 85MHz

Sun custom low-power CMOS integrated ASICs for high reliability:

- microSPARC II (SPARC Version 8 compatible RISC microprocessor)
  - SPARC IU
  - SPARC FPU
  - Cache controller
  - Instruction cache (16 KB organized as 512 32-byte blocks)
  - Data cache (8 KB organized as 512 16-byte blocks)
  - Reference MMU (64-entry fully associative TLB)
  - I/O MMU
  - DRAM controller (main memory)
  - SBus controller
  - JTAG internal/boundary scan controller
- MACIO (master I/O controller)
  - DMA2 controller
  - SCSI-2 (ESP) controller
  - Ethernet controller (LANCE-compatible)
  - Parallel port controller (Centronics compatible)
  - JTAG internal/boundary scan controller
- SLAVIO (slave I/O controller)
  - Serial A and B controller (synchronous/asynchronous)
  - Floppy disk controller
  - Interrupt controller
  - Reset controller
  - Power on/off controller
  - Counter/timer controls
  - EPROM/NVRAM controller
  - JTAG internal/boundary scan controller

- Keyboard and mouse (or serial port C and D)
- System LED

Components from other vendors:

- SC64 VMEbus Controller
  - A64/32/24/16 master capabilities
  - A64/32/24 slave
  - D64/32/16/8 data transfer widths
  - DMA controller
  - Four level bus arbiter
  - Block mode transfer support
  - Location monitor interrupt
  - Sysfail LED
- KSIC Interface
- Socketed 4Mbit (512Kx8) boot PROM (OBP)
- Two 256/512KB Flash EPROM
- 8K TOD/NVRAM (with battery backup)
- Front panel rotary switch
- Four user LEDs
- Reset/Abort toggle switch

Connectors

- Two or four SIMM connectors
  - supports 4/ 16 MB SIMMs or double density 8M/32MB SIMMs at 60 ns
  - up to 160 MB of on-board main memory expansion
  - one parity bit per 32-bit word
  - 64-bit memory bus
- Two SBus master/slave connectors
  - 32-bit data path
  - 32-bit virtual address for master, 28-bit physical slave address
  - Maximum 16 byte burst
- High density SCSI-2 connector
- Sun keyboard/mouse connector
- 10BaseT Ethernet connector

- AUI Ethernet connector
- Serial port A/B connector

2.2 *Themis SPARC 5/64 Supported Standards*

- Fully compatible with SunOS 5.4 and Solaris 2.4 (or higher)
  - Binary compatible with SPARCware applications (Solaris 1.0 or later)
  - Disk I/O read and write clustering
- Complete implementation of SunOS 4.1.3 and Solaris 1.1.2
  - Binary compatible with sun4m architecture
  - Runs all Solaris 1.1.2 applications
- Hardware standards compliant:
  - SPARC International SPARC Compliance Definition 1.0
- Software standards compliant:
  - IEEE POSIX standard 1003.1 (1988)
  - X/Open-XPG3 component branded
  - XPG2 BASE branded
  - System V Interface Definition Issue II
- Supports standards in application software:
  - SPARC standards
  - OpenLook standards
  - Sun UNIX standards
- 8000+ SPARCware applications

Themis SPARC 5/64 Description

The Themis SPARC 5/64 is a SPARC RISC (Reduced Instruction Set Computer) based VMEbus board.

Fully compatible with the SUN SPARCstation 5 designs, the 5/64 SBC (Single Board Computer) features the Fujitsu / Sun microSPARC II microprocessor, a master I/O (MACIO) controller, a slave I/O (SLAVIO) controller, SIMM DRAM (main memory) and a Themis custom KSIC interface to the SCV64 VMEbus controller.

On-board peripheral connectors include a full VME A64D64 interface, two SBus slots, SCSI-2, 10BaseT and AUI Ethernet, keyboard/mouse, parallel printer port/floppy disk interface and two serial ports (A & B).

Most peripheral signals are also available on the user pins of the VME P2 connector. A paddle board provides industry standard connectors for AUI Ethernet, Centronics parallel (or floppy disk, user selectable), SCSI-2 and two serial ports (C & D) to the P2 I/O interfaces.

Figure 3-1 Layout of the Themis SPARC 5/64 Board

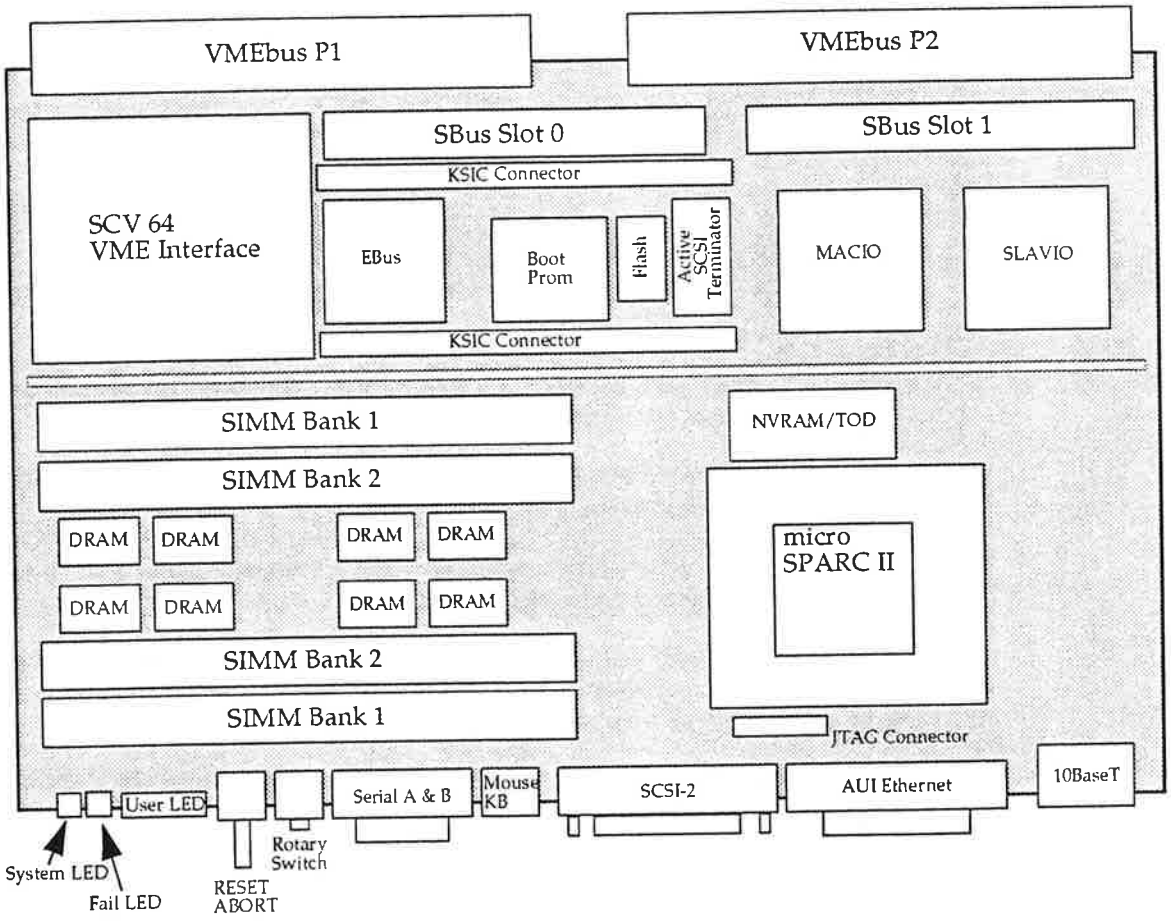
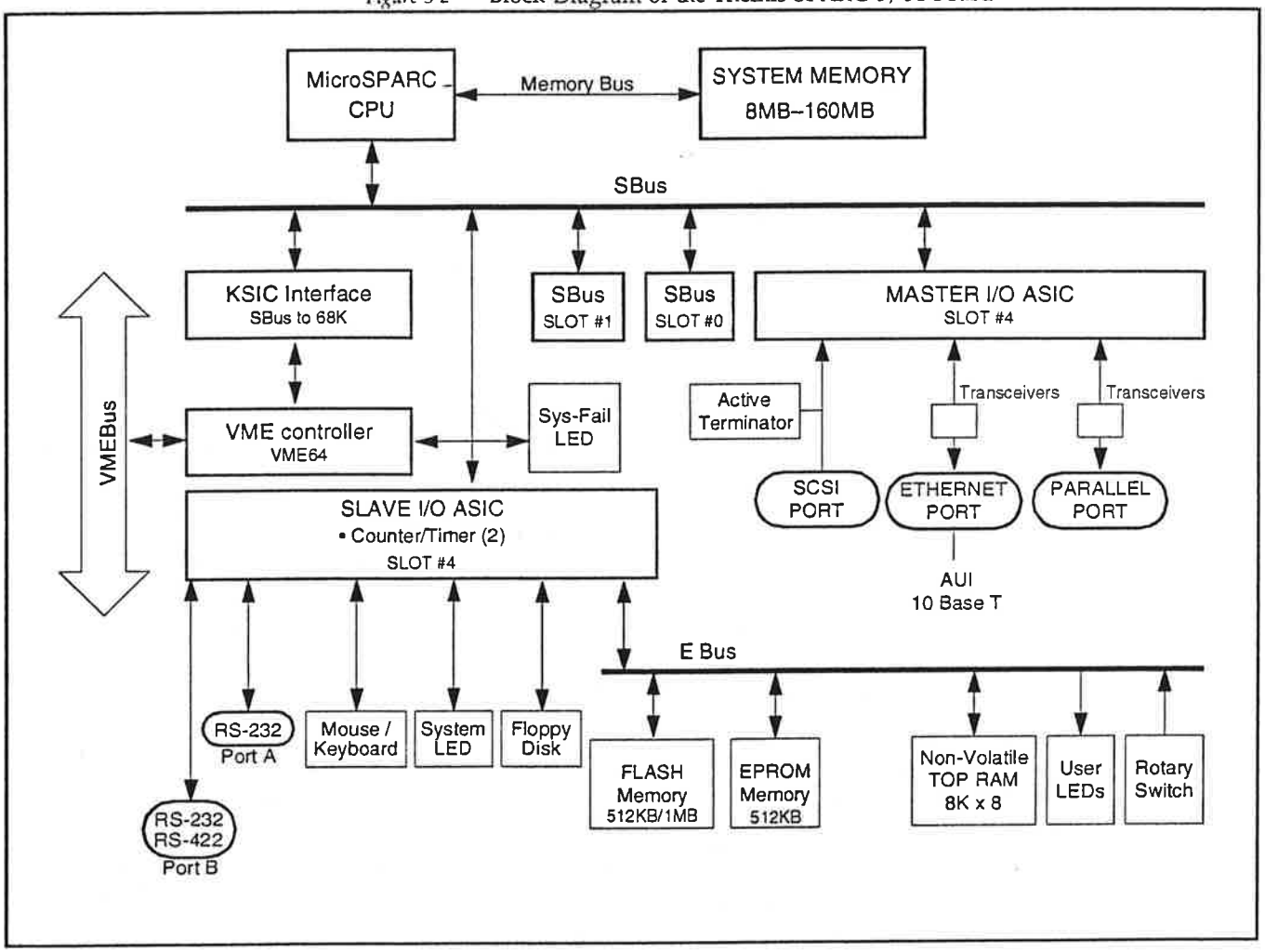


Figure 3-2 Block Diagram of the Themis SPARC 5/64 Board



3.1 Themis SPARC 5/64 Main Components

3.1.1 microSPARC II Microprocessor

microSPARC II is a single-chip microprocessor that contains an Integer Unit (IU), a Floating Point Unit (FPU), data and instruction caches, and controllers for Main Memory Cache, Random-Access Memory (RAM), and Memory Management Unit (MMU).

The microSPARC II processor runs at 85MHz, and 110MHz, with a processing capability of 112.5 million instructions per second (MIPS) @ 85MHz. The FPU runs at 14.9 million floating-point operations per second (MFLOPS)[1] @ 85MHz.

Integer Unit (IU) and Floating-Point Unit (FPU)

The core of the microSPARC II is the combined SPARC Integer Unit/Floating-Point Unit (IU/FPU).

The IU includes a 32-bit external bus interface with separate data and address buses, a five-stage instruction pipeline, a barrel shifter, two data aligners, and a three-port register file consisting of 128 registers. These registers are configured into overlapping sets that facilitate the passing of parameters. All instructions with the exception of double loads, stores, and floating-point operations are executed in one machine cycle.

Main Memory Cache

Cache memory included on the microSPARC II chip provides high-speed local memory for the IU. The 8 KB data cache is a direct-mapped physical-address write-through cache with no write allocate, organized as 512 lines of 16 bytes. The 16KB instruction cache is a physically-addressed cache, organized as 512 lines of 32 bytes.

1. Performance measurement specifications are defined in chapter 3

Reference Memory Management Unit (MMU)

Translation of 32-bit virtual addresses from the CPU is performed by a SPARC Reference MMU as defined in the SPARC Architecture Manual Version 8. The microSPARC II MMU also functions as an I/O MMU. It controls arbitration between I/O, data cache, instruction cache and TLB references to memory.

The SBus is the basic communication mechanism between the processing core (CPU), the MMU, main memory and various I/O devices. The MMU translates the virtual address output of the CPU and drives the resulting physical address on the SBus. It also decodes this physical address into a set of select signals used to enable main memory or I/O devices on the SBus.

Translation Lookaside Buffer (TLB)

The TLB is 32-entry fully-associative reference translation-lookaside buffer that uses a pseudo-random algorithm for the replacement of TLB entries, all contained within the MMU.

RAM Controller

The Dynamic Random Access Memory (DRAM) controller provides all control signals to support and address up to 160MB of onboard DRAM. It supports 60 ns DRAMs to give burst reads and writes at one word per clock. It also provides parity generation and checking. It has a 2-word write buffer to provide quick response to cache write-through operations. The DRAM controller includes a private RAM data bus, isolating the DRAM data from the SBus data. This lowers the load on the SBus data lines, in addition to allowing the RAM controller to do buffered writes while releasing the SBus.

SBus Controller

The microSPARC II CPU controls the 21.25 MHz SBus (85MHz), and 22MHz SBus (110MHz) and all SBus devices in the three vitual SBus slots, and in the two physical single-height SBus slots.

Table 3-1 Themis SPARC 5/64 SBus Slots

SBus Slot Number	Function of SBus Slot
Slot 0	Physical SBus slot 0 for an SBus Card or VME access or combination thereof
Slot 1	Physical SBus slot 1 for an SBus Card or VME access or combination thereof
Slot 2	VME port
Slot 3	
Slot 4	SLAVIO and MACIO I/O controllers, VME MMU, VME port and SCV64 controller

Table 3-2 microSPARC II General Mapping

SBus Slot 4	256 Mbytes	0x7000.0000
SBus Slot 3	256 Mbytes	0x6000.0000
SBus Slot 2	256 Mbytes	0x5000.0000
SBus Slot 1	256 Mbytes	0x4000.0000
SBus Slot 0	256 Mbytes	0x3000.0000
Control Spage	256 Mbytes	0x2000.0000
Unused	128 Mbytes	0x1000.0000
DRAM bank 7	32 Mbytes	0x0E00.0000
DRAM bank 6	32 Mbytes	0x0C00.0000
DRAM bank 5	32 Mbytes	0x0A00.0000
DRAM bank 4	32 Mbytes	0x0800.0000
DRAM bank 3	32 Mbytes	0x0600.0000
DRAM bank 2	32 Mbytes	0x0400.0000
DRAM bank 1	32Mbytes	0x0200.0000
DRAM bank 0	32Mbytes	0x0000.0000

3.1.2 *Input/Output Controllers and 8-Bit Devices*

MACIO (Master I/O Controller)

The MACIO ASIC implements DMA driven input/output interfaces on the Themis 5/64. Its functionality is divided into a set of macrocell based sections dedicated to a particular device.

Ethernet Controller Macrocell

Ethernet is supported via a LANCE compatible controller. A 10BaseT and AUI interface are available at the front panel. The AUI interface is also routed to the P2 connector and accessible via the paddle board.

SCSI-2 Controller Macrocell

SCSI device operations are handled through a SCSI-2 controller macrocell. All SCSI bus transfers are performed by the on-chip DMA controller. The controller provides transfer speeds of up to 10MB/s.

DMA Controller Macrocell

Direct-Memory Access (DMA) is handled through the DMA2 controller macrocell. It includes three external interfaces designed to provide DMA access to Ethernet, SCSI, and the bi-directional Centronics parallel port.

To minimize SBus usage, the DMA2 controller has internal line buffers and performs SBus burst operations of 16 bytes where possible.

The DVMA (Direct Virtual Memory Access) function of the DMA2 controller provides two channels of direct memory access between the main memory on the SBus and the SCSI and Ethernet interfaces.

DVMA also provides the data path between the SBus and the SCSI and Ethernet interface devices required for the IU to initialize and configure these interfaces.

Printer Port Controller Macrocell

Centronics-compatible bi-directional parallel port operations are handled through the DMA2 macrocell.

SLAVIO (Slave I/O Controller)

All non-DMA and 8-bit interface connections are implemented by the SLAVIO ASIC controller.

Serial Port A Controller

Synchronous or asynchronous RS-232 or RS-422 interface with modem control lines and baud rates up to 38400. Channel A interface includes: RxD, RxClk, DCD, CTS (Inputs) and TxD, TxCLK, DTR, RTS (Outputs).

Serial Port B Controller

Synchronous or asynchronous RS-232 or RS-422 interface (selectable through solder beads IC3008A and IC3008B) with modem control and baud rates up to 38400. Channel B interface includes: RxD, RxClk, DCD, CTS (Inputs) and TxD, TxCLK, DTR, RTS (Outputs).

Floppy Disk Controller

Interface to standard, SUN compatible floppy disk device. This port is only available on the P2 connector (paddle board) and occupies the same pins as the Centronics compatible printer interface. Solder bead SB5600 determines parallel port or floppy controller access on P2 bus. Factory standard configuration is parallel port access.

Keyboard/Mouse Controller

Standard Sun Microsystems keyboard/mouse ports are provided to connect a keyboard and a mouse through the standard 8-pin connector located on the front panel TTL level) or via the serial C&D ports on the P2 (RS-232 level) with TxD and RxD signals only.

Note: Power to the keyboard/mouse set is provided through the 8-pin DIN connector and protected against short circuit with a polyswitch.

EBus Controller

The EBus is a SLAVIO 8-bit sub-bus for controlling low speed peripherals.

The OBP PROM, NVRAM/TOD and Flash EPROM components are attached to the SLAVIO ASIC through the external EBUS 8-bit interface.

JTAG Internal & Boundary Scan Controller

Signal paths and circuitry are provided to perform a ASIC-level diagnostic examination of SLAVIO.

Interrupt Controls

SLAVIO contains a uni-processor subset of the Sun-4M interrupt controller. It includes all system logic to support 15 levels of software and/or externally generated interrupts. Assignment and prioritizing of these interrupts is performed by the interrupt logic.

Table 3-3 Interrupt Level Assignments.

CPU Level	Source	SBus Source	Device
0	No Interrupt pending		
1	SOFT Intr 1		
2	SOFT Intr 2	Intr 1	Parallel Port
3	SOFT Intr 3	Intr 2	
4	SOFT Intr 4		SCSI
5	SOFT Intr 5	Intr 3	
6	SOFT Intr 6		
7	SOFT Intr 7	Intr 4	
8	SOFT Intr 8		
9	SOFT Intr 9	Intr 5	
A	SOFT Intr 10		System Counter/Timer
B	SOFT Intr 11	Intr 6	Floppy
C	SOFT Intr 12		Keyboard/Mouse - Serial Ports
D	SOFT Intr 13	Intr 7	
E	SOFT Intr 14		Processor Counter/Timer
F	SOFT Intr 15		Asynchronous H/W Errors

System Reset Controls

Three reset sources are recognized by the system reset controller: power-on reset (POR), software reset (SWR) and the front panel reset switch. Either of these sources will cause SLAVIO to assert a reset signal, returning the Themis SPARC 5/64 board to a known state.

Counter/Timer Controls

SLAVIO features two programmable counter/timer channels.

SCV64 VMEbus Controller

The Newbridge SCV64 advanced VMEbus controller is used to implement the A64D64 VMEbus interface. Themis designed the KSIC interface logic (see below) to glue the SCV64 controller to the SBus logic of the SPARC design.

A flexible VMEbus memory map implementation allows variable configuration of the available SBus slot spaces to either the VMEbus, SBus or a combination thereof.

Each 256MB SBus slot is subdivided into 16MB sections which can be individually mapped to specific SBus or VMEbus addresses, preserving the flexible addressing scheme (addressing modes, etc.) of the VMEbus.

KSIC SBus to SCV64 Interface

To interface the 68K oriented physical interface of the NewBridge SCV64 VMEbus controller to the SBus based logic of the microSPARC II SPARC processor, Themis designed a flexible KSIC interface logic, implemented on a daughterboard design on the 5/64 motherboard.

Slot 0: Extension SBus Slot

Standard SBus slot with master/slave capability. It provides a 256MB add-on SBus card space associated with connector J601 on the processor board. The 256 MB address space of this slot is divided into 16 slots of 16MB each. Each individual 16MB section can be allocated as SBus or VMEbus memory through the use of the VME MMU address translation table.

Caution – This slot in unavailable on an 5/64 board with 4 occupied SIMM sockets installed.

Slot 1: Extension SBus Slot

Standard SBus slot with master/slave capability. It provides a 256MB add-on SBus card space associated with connector J602 on the processor board. The 256MB address space of this slot is divided into 16 slots of 16MB each. Each individual 16MB section can be allocated as SBus or VMEbus memory through the use of the VME MMU address translation table.

Single Inline Memory Modules (DRAM SIMMs)

4/8/ 16 or 32 MB SIMMs can be used in the two or four[1] 5/64 SIMM sockets. Main memory can be expanded up to 160 MB. The two SIMM socket configuration allows physical access to both SBus connectors, while the four SIMM socket version prevents mechanical access to the SBus 0 connector.

The 5/64 is designed to use 4/8/16 or 32MB 60 ns Sun Microsystems 33 bit SIMMs that support a single parity bit per 32 bit word. Standard SIMMs can also be used with the 5/64; three of the four parity bits remain unused. The SIMM sockets #1 support 4MB, 16MB and 32MB of Sun compatible memory

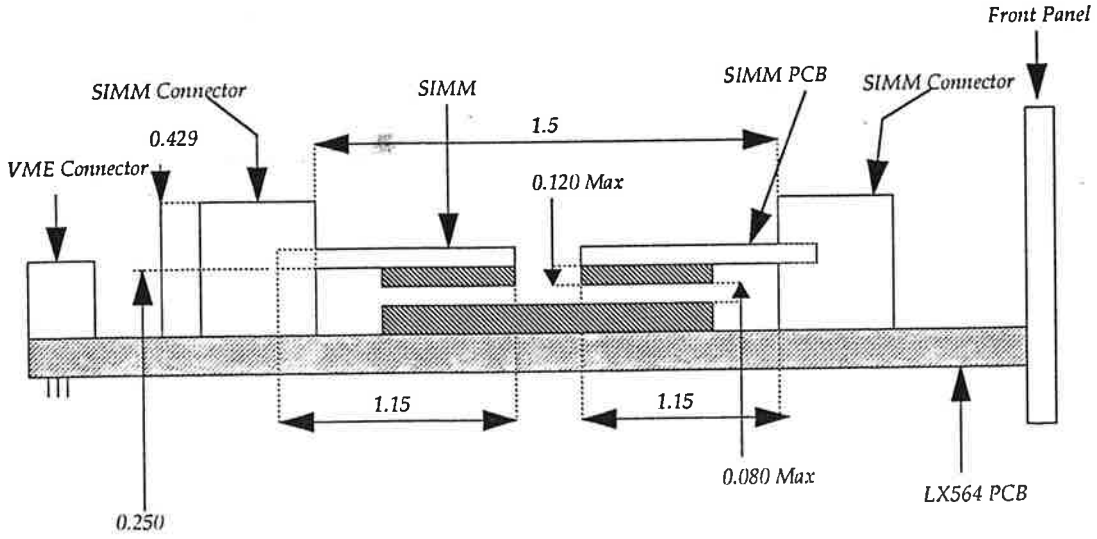
1. Factory configuration required

modules. The horizontal SIMM socket used on the SPARC 5/64 board is AMP horizontal SIMM connector, 7-382486-2. The approved vendors list for double, and single SIMMs is listed below:

Table 3-4 Approved list of vendors

Vendor Name	Vendor Address	Vendor Telephone #	SIMM #
CENTON Electronics	1731 Technology Drive, Suite 570, San Jose, CA 95110	(408) 441-1200	16MB SIMM Modules: CUN 16M / SPARCLX (2x 8MB CUN 8M / SPARCLX (2x 4MB) CUN 32M / SPARCLX (2x 16MB)
Kingston Technology Corp.	17600 Newhope St. Fountain Valley, CA 92708	(714) 435-2600	KTS-4000 / LX

Figure 3-3 SPARC 5/64 board - SIMM diagram



Open Boot PROM (OBP)

The Open Boot PROM is connected to the Sun Microsystems Expansion Bus (EBus)_OBP is contained in a 512Kx8 PROM (TI27C040-90ns PLCC32) providing the following functionality:

- Runs start-up diagnostic tests.
- Initializes the host machine.
- Reads non-volatile RAM (NVRAM) and executes the boot sequence, typically booting SunOS. The Diagnostic Executive or standalone programs can also be executed.
- Includes the abbreviated system monitor. Entry into the system monitor is signified by the > prompt. If a boot attempt fails, the Open Boot PROM tries to start the abbreviated system monitor.
- Supplies program code for the FORTH Toolkit; the on-board diagnostics containing the FORTH Toolkit and the FORTH language interpreter. Entry into the FORTH Toolkit is signified by the ok prompt.
- VMEbus related board settings are controlled by a number of environment variables and commands.

Non-volatile RAM (NVRAM)

8 KB of Non-Volatile Random Access Memory (NVRAM) are provided by the TOD (time-of-day) chip connected to the SLAVIO EBus. A single lithium battery within the NVRAM module provides battery backup for approximately 10 years.

The NVRAM stores the default system configuration parameters. You can modify these parameters using the FORTH Toolkit in the OBP.

Time-of-Day Clock (TOD)

The NVRAM chip contains the time-of-day clock (TOD). A lithium battery within the NVRAM module provides battery backup for the TOD.

FLASH EPROM

Two soldered FLASH EPROMs with 512KB capacity each. The FLASH EPROM reside in the 16MB address space of the BOOT ROM and the allocation of EPROM and FLASH EPROM, as the boot device, is rotary switch dependent. For more information regarding allocation via rotary switch, please refer to Chapter 5, *OpenBoot PROM Configuration on page 5-38.*

3.1.3 *5/64 Board Connectors*

SBus Slot (SBus Devices) Connectors

Two physical SBus 96-pin high-density connectors are located on the Themis SPARC 5/64 board supporting either 2 single slot or one dual slot SBus card(s)[1].

The SBus is an open systems 32-bit synchronous bus with a 28-bit address space and support any available SBus board. The physical slots have full master/slave capabilities. Additional mechanical support must be provided in your enclosure design for the SBus cards, as defined in the SBus specifications.

Main Memory DRAM SIMM Female Connectors

Two or four SIMM sockets allow memory expansion by up to 64MB in addition to the 8 or 32MB of onboard DRAM. The physical characteristics of the four socket design prevents access to the first SBus slot (SBus 0).

SCSI Female Connector

This front panel, 50-pin dual-row high density SCSI-2 connector can be used to attach any SCSI/SCSI-2 device. All SCSI signals are also provided on the backplane P2 connector and are accessible through the paddle adapter.

---

1. Single slot SBus card only on 4 SIMM socket version.

10base-T Ethernet

Supplies Ethernet connectivity to a hub or router (standard RJ45 connector).

AUI Ethernet

Standard DB-15 front panel AUI Ethernet connector. All AUI signals are also provided on the P2 connector and accessible through a standard AUI connector mounted on a paddle board.

Mouse/Keyboard Connector

Support for a standard SUN Type-4 or Type-5 keyboard and mouse.

Serial Port A&B

High density DB-26 front panel connector for serial ports A and B.

3.2 System Considerations

3.2.1 Required Initial Power-Up System Components

A VME system rack with power supply

Table 3-5 DC voltage as specified in the VME Specs.

Supply Voltage	Minimum	Maximum
+5VDC	+ 4.875V	+5.25V
+12	+11.64V	+12.6V
-12	-12.6V	-11.64V

Themis SPARC 5/64 Specifications

4

4.1 Themis SPARC 5/64 Board

Physical Dimensions	
CPU Board Size:	6U VME form factor
Height	233.35 mm (9.187 in.)
Depth	160 mm (6.299 in.)

Microprocessor Performance Figures (85MHz)	
IU Performance	112.5 MIPS (Dhrystone)
FPU Performance	14.9 Linpack 100 MFLOPS (with Apogee compiler)
SPECint92	64
SPECfp92	54.6

RISC Microprocessor	
Microprocessor	microSPARC II @ 85MHz / 110MHz
Cache Controller	Cache
Memory Controller	Reference MMU
Testing Support	JTAG scan support, built-in self-testing
Packaging	321 pin PGA with Heatsink

RISC Microprocessor (Continued)	
Integer Unit	Sun Microsystems SPARC IU
IU Performance	112.5 MIPS @ 85MHz
Architecture	Harvard
Cache Streaming Support	Instruction and Data
Floating Point Unit	Meiko Design FPU
FPU Performance	14.9 MFLOPS @ 85MHz
Instructions	Single and double precision
Cache Instruction	16 KB direct-mapped write-through, virtual
Cache Data	8 KB direct-mapped write-through, virtual
Cache Write Buffering	Two double-word
Cache Recovery	Complete write-fault recovery supported
Cache Linesize	32-byte (I) -16-byte (D)

Input/Output	
SBus Devices	5 master/slave devices
SBus Slots	2 physical slots
SBus Standard	SBus specifications, Revision B.0
SBus Data Size	32-bit full master/slave (all slots)
SBus Clock Rate	21.25 @ 85MHz
SBus Write Sizes	Maximum 16 byte burst
SBus Writes	Quad/doubleword burst mode supported
SCSI-2 Port	One 50-pin connector and signals to backplane P2 connector
SCSI Standard	ANSI X3.131
SCSI-2 Types	synchronous/asynchronous (host)
SCSI-2 Performance	Up to 10 MB per second (synchronous) (fast)
Ethernet Port	Twisted-pair signals to RJ45 front connector; AUI signals to backplane P2 connector and front panel
Ethernet Standard	IEEE 802.3
Parallel Port	Signals to backplane P2 connector
Parallel Port Standards	Bi-directional 8-bit Centronics-compatible

Input/Output (Continued)	
Serial A/B Ports	Signals to DB 26-pin front connector
Serial Standard	RS-232C (A & B) or RS-422 (B only)
Serial Types	synchronous/asynchronous
Serial Performance	38400 (38.4 Kbit) bauds
Control Lines	RTS, DTR (output) CTS, DCD (input)
Clock Lines	TXC (Output), RXC (Input)
Serial C/D Ports*	Signals to backplane P2 connector
Serial Standard	RS232
Serial Types	asynchronous
Serial Performance	38400 baud
K/M Port*	Signals to 8-pin DIN front connector
K/M Type	asynchronous, TTL level
K/M Performance	SUN compatible
Floppy Disk Controller	SUN compatible floppy disk interface with signals available on P2 and paddle board. Floppy interface or parallel printer port are selectable by solder beads. Factory standard configuration if parallel port active.

* Keyboard/Mouse ports and serial ports C/D cannot be used simultaneously.

On-board Main Memory	
Type of memory	Parity memory (1 parity bit per 32 bits)
On-card DRAM Banks	2 or 4
SIMMs per Bank	2
DRAM SIMM Specs	72-pin (x 33 or x 36)
DRAM Speed	60 ns fast-page mode
DRAM Expansion Range	8 MB to 160 MB (16 MB minimum for Solaris 2.x)

SBus	
SBus Controller Chip	microSPARC II
SBus Physical Ports	Two female SBus 96-pin connectors

VMEbus	
VME Interface Controller	Newbridge SCV64 and Themis KSIC interface
VME System Controller	Multiple-level arbiter
	Round-robin or Full priority
	Arbitration Time-out
	Data transfer time-out
VME Master Interface	CPU or DMA sourced transfer
	A64/32/A24/A16 addressing mode
	D64/D32/D16/D08 (E0) transfer mode
	Master BLT and MBLT supported (with DMA)
	Requester options (unconditional or Fair mode)
	Release options (ROR, RWD, BCLR)
VME Slave Interface	A64/A32/A24 addressing mode
	D64/D32/D16/D08 (E0) Transfer mode
	Slave BLT and MBLT supported
	Programmable slave base address
	Programmable slave window size
	Location monitor
Interrupter	8-bit vector
	IRQ 1-7 (One at a time)
Interrupt Handler	IH (1-7) 8-bit vector
	IACK daisy-chain driver
	SCV64 to SBus interrupt level translation

4.2 Themis SPARC 5/64 Electrical Specifications

Power Requirements For This Product	
+5V:	+5VDC ± 5% (4.75V min./5.25V max.)
Configuration with 32Mbytes:	
	typical: 2.8A
	max: 3.8A
SBus Requirements:	min: 4.75V
	max: 5.25V
	2A max (possibly 3A during 1ms)
+12V: *Not required when no AUI and SBus card installed*	
SBus Requirements:	min: 11.25V
	max: 12.75V
	30mA max (possibly 50mA during 1ms)
MAU Requirements:	400mA max (protected by Polyswitch)
+12V: *Not required when no AUI and SBus card installed*	
SBus Requirements:	min. -12.75V
	max: -11.25V
	30mA max (possibly 50mA during 1ms)

4.3 *Themis SPARC 5/64 Environmental Specifications*

The following environmental specifications for the Themis SPARC 5/64 board are presented below.

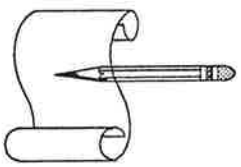
Operating Environmental Requirements For This Product		
Values	*Minimum*	*Maximum*
Temperature Range	32° F (0° C)	122° F (50° C)
Humidity Range relative non-condensing at 104° F (40° C):	5%	95%
Altitude Range	0 feet (0 m)	30,000 feet (9,144 m)

Non-Operating Environmental Requirements For This Product		
Temperature Range	-4° F (-20° C)	167° F (75° C)
Humidity Range relative non-condensing at 104° F (40° C):	5%	95%
Altitude Range	0 feet (0 m)	40,000 feet (12,192 m)

Air Flow Requirements For This Product		
At 85Mhz	100 LF/m	@ 50° C
At 110Mhz	220 LF/m	@ 50° C

Themis SPARC 5/64 Installation

5.1 Registering Your SPARC 5/64



Please review the Themis Computer warranty and complete the product registration card delivered with your 5/64 board(s). Return of the registration card is not required to activate your product warranty, but by registering your 5/64, Themis Computer we will be able better provide you with timely update information and product enhancement notifications.

At Themis Computer we value our customers comments and concerns. This is why we have a marketing department who is eager to know what you think of our products, along with a customer support department that is committed to providing the best product support in the industry unsurpassed.

Customer support is available 8AM to 5PM PST, Monday through Friday via telephone, fax or email. You can also visit our web site.

Customer Support:
Phone: 510-252-0870
Fax: 510-490-5529
Email: support@themis.com
Web Site: www.themis.com

5.2 *Unpacking The SPARC 5/64*

Caution – The SPARC 5/64 contains statically sensitive components. Industry standard anti-static measures must be observed when removing the 5/64 from its shipping container and during any subsequent handling

A *wrist strap* (or *grounding strap*) provides grounding for static electricity between your body and the chassis of the system unit. Electric current and voltage do not pass through the wrist strap.

Before handling the 5/64, attach the wrist strap to your wrist and to the metal casing of your grounded chassis.

Caution – Boards and modules can be damaged by harmful electrical charges if you do not wear a wrist strap.

To attach the wrist strap:

1. **Wrap the grounding strap twice around your wrist.**
   Make sure the adhesive side is against your skin
2. **Attach the grounding wire to a chassis or other ground connection**

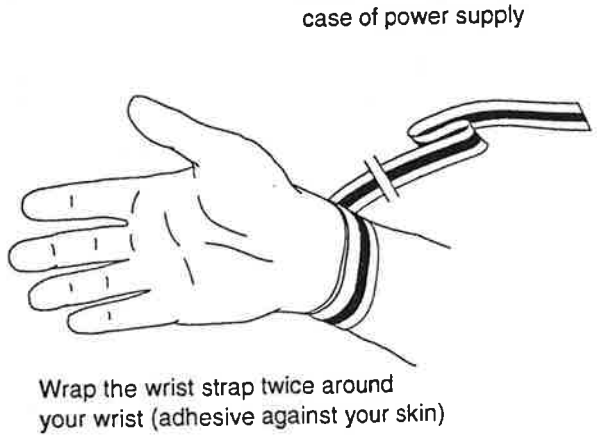


Figure 5-1 Attaching a Wrist Strap

Remove the 5/64 and accessories from its shipping container and check the contents against the packing list. The package should include:

1. SPARC 5/64
2. Integration Kit, (if ordered)
3. This manual (if ordered)
4. Dual serial null-modem cable
5. SCSI cables (2)
6. 5/64 paddle board

Report any discrepancies to Themis customer support immediately.

Remove the SPARC 5/64 CPU board from its anti-static wrapping. If you have ordered your 5/64 with more than 32MB DRAM, 2 or 4 SIMM sockets will be occupied by SIMM modules.

5.3 *Configuring the SPARC 5/64*

Confirm installation of all factory default jumpers (see diagram below and *5/64 Jumper Settings on page G-123*). The default configuration is as follows:

- 5/64 as Slot 1 System Controller
- Keyboard/Mouse ports connect to front panel
- Active SCSI termination enabled
- Front panel RESET switch enabled
- Parallel printer port available on P2
- VMEBus BERR signal connected to SCV64
- Serial port B is RS232
- VME interface enabled
- FLASH programming disabled

If the default jumper settings meet your requirements you are now ready to install the 5/64 in a standard VME chassis. If re-configuration is required, follow the instructions in *5/64 Jumper Settings on page G-123* for setting board jumpers to match your application.

Caution – If you are not installing the 5/64 as a slot 1 system controller, jumper J2001 must be set to 2-3.

Figure 5-2 5/64 default jumper settings (solder side)

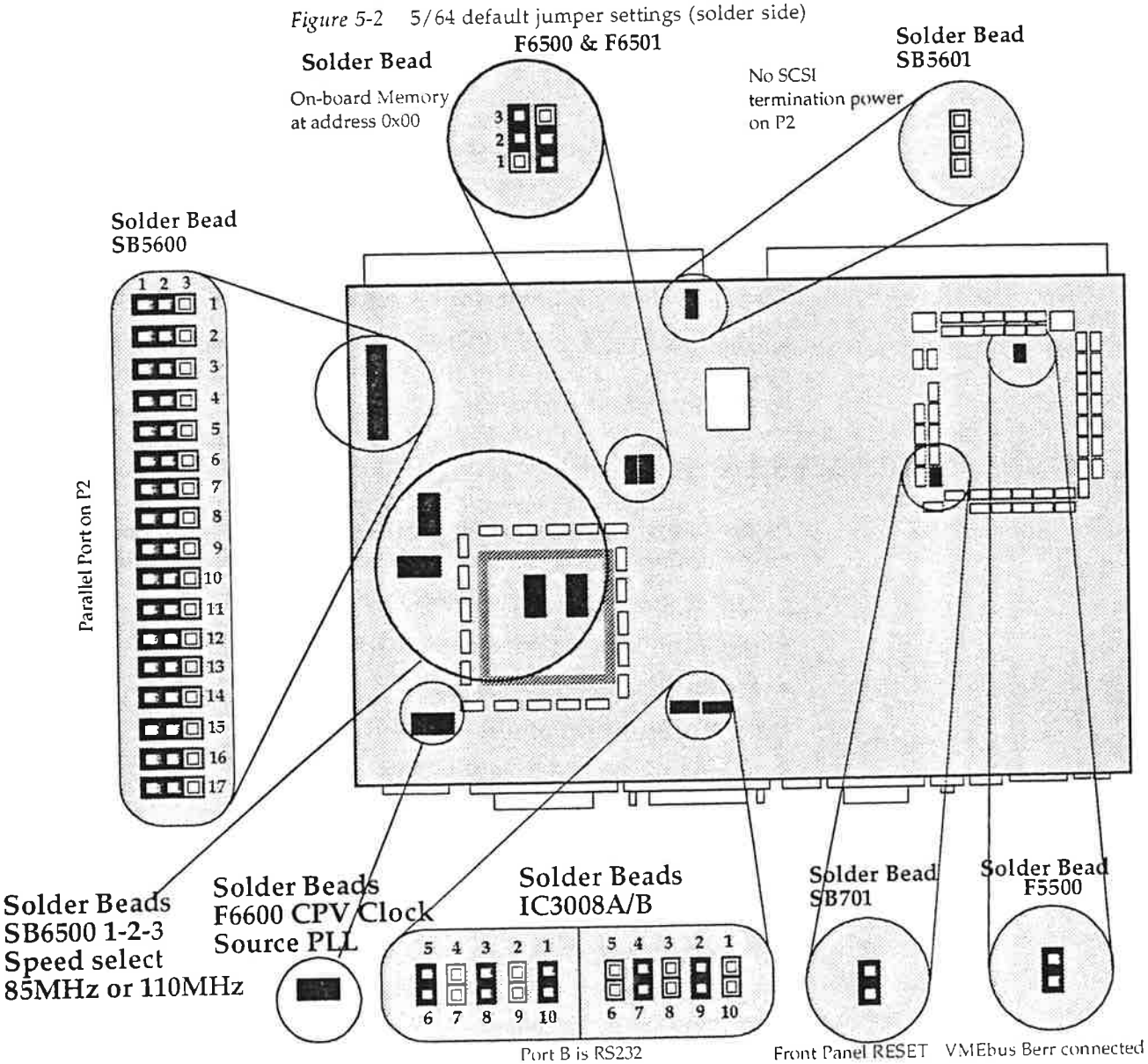
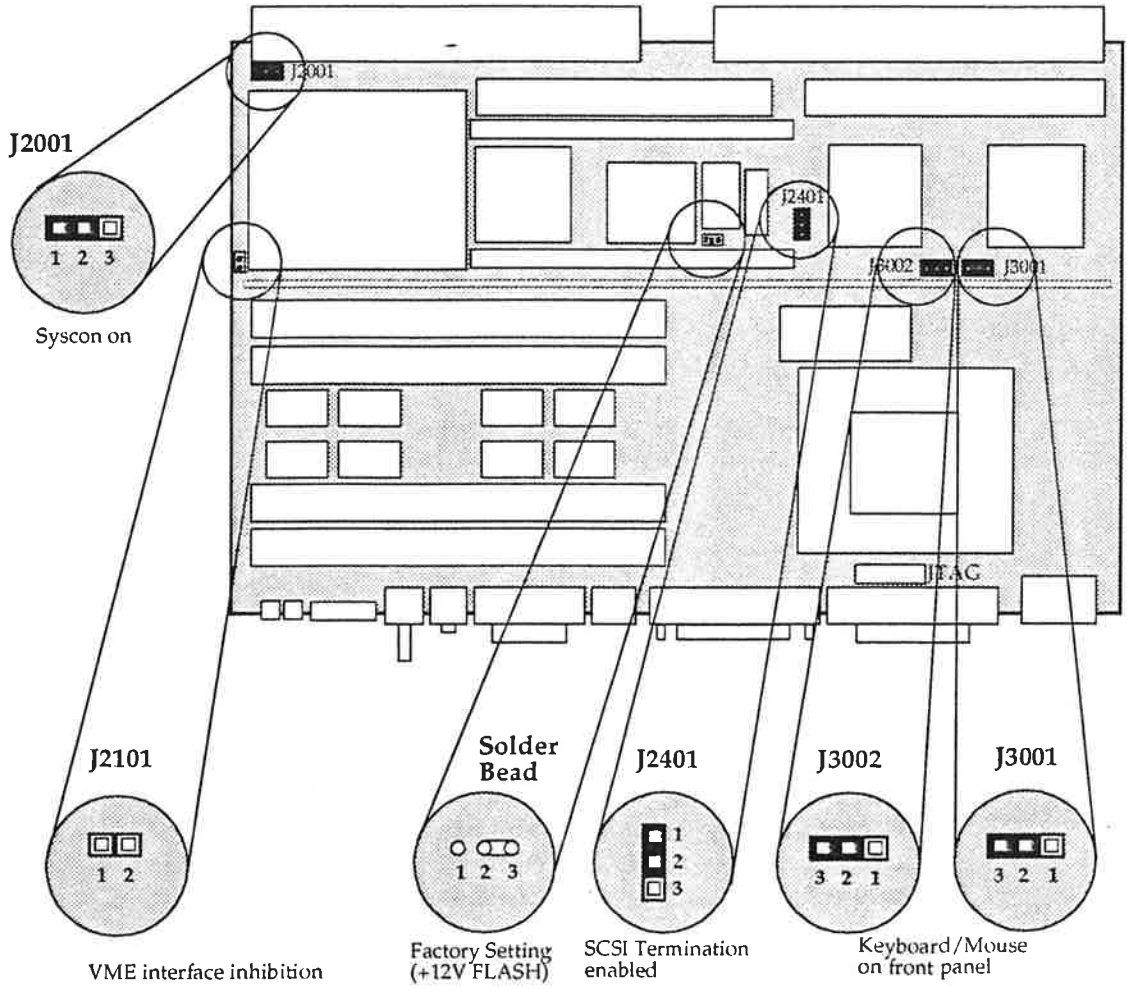
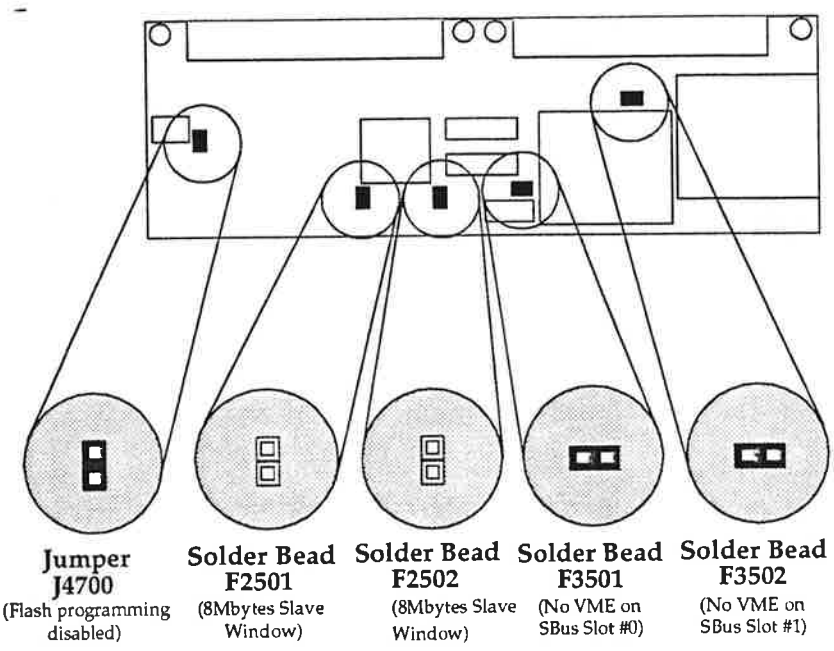


Figure 5-3 5/64 default jumper setting (component side)



In addition to the 5/64 hardware a standard VME chassis with P1/P2 backplane is required. If you intend to use the 5/64 in a workstation configuration instead of as an embedded controller, you will also need a hard disk and graphics frame buffer or serial terminal.

Figure 5-4 KSIC Interface default settings



5.4 *OpenBOOT PROM Configuration*

The BOOT PROM is physically composed of three PROM devices. Two are FLASH EPROMs (two 512KB) and one is a single, socketed UVPROM (512KB) typically containing the OpenBoot PROM code or operating system specific boot code (such as the VxWorks bootrom).

At power-up, the processor fetches instructions starting at one physical address 0x7000.0000. The user may locate either the UVPROM or one of the FLASH EPROM devices at this location to serve as the boot device. This feature allows the user to conveniently debug boot code or add and use their custom boot code.

When mapping #2 is selected. Flash #1 is still programmable (with the extend Jumper 4700 is set appropriately) though it is the boot device.

In order to program the flash, the following OBP commands can be used:

syntax: flash-copy (addr size n --)

This command allows the user to transfer a contiguous memory block to the selected Flash device.

addr is the start virtual memory address

size is the memory block size to be programmed starting at the location addr.

n is the Flash number (1 or 2)

Note that the start address on the destination (which is a Flash device) is its base address.

Application example:

ok 4000 dload my-flash-image (download code from a server)

ok 4000 40000 1 flash-copy (copy it (256k) into Flash #1)

Does not work

syntax: flash-erase (n --)

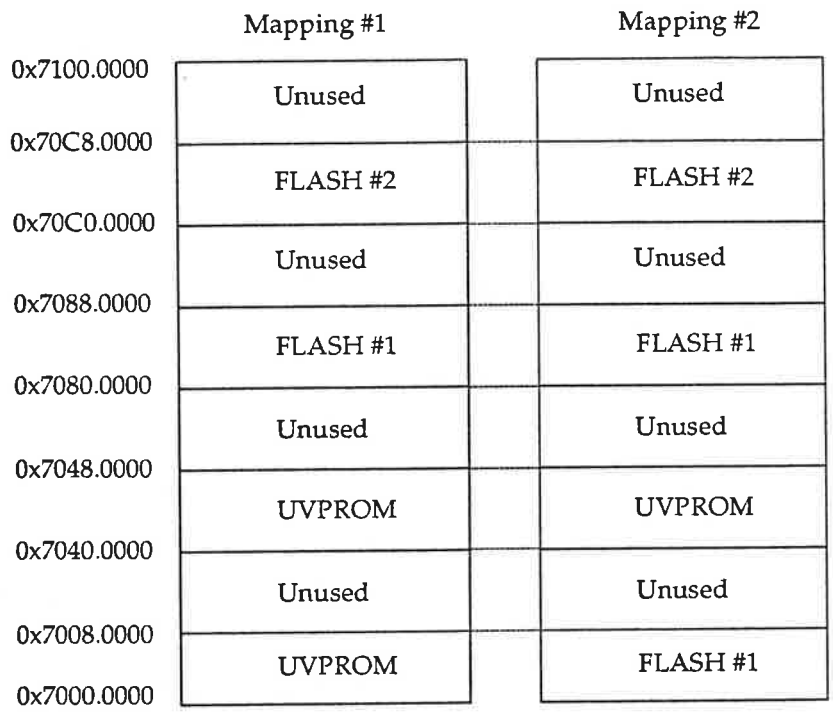
This command allows the user to blank the selected Flash memory device.

n is the flash number (#1 or 2)

Note: After this line type and you hit Enter/Return it first does a "Flash Erase" then copies

The mapping of the three PROM devices within the BOOT PROM address map is determined by the setting of the front panel rotary switch. A rotary switch value of 0 through 7 selects mapping #2 and a value of 8 through 15 selects mapping #1 as shown below.

Figure 5-5 BOOT PROM Mappings



Caution – The rotary switch setting is only latched when the board reset of the 5/64 is released. Modifying the rotary switch while the 5/64 is operating has no effect until a board reset occurs. However it is mandated not to change its setting while the 5/64 is operating.

5.5 *Configuring The VME Interface*

Themis has implemented a variable and flexible VMEbus interface using both on-board jumpers and OpenBoot PROM commands and environment variables specific to the 5/64 board.

The 5/64 is typically re-configured when VMEbus boards are added, removed or changed in the VMEbus chassis. Board configuration normally involves allocation of VMEbus master access address, interrupts and slave base address of the 5/64.

A small number of VME interface capabilities are configured using jumpers or solder beads, including:

- The slot-1/system controller capability.
- VMEbus isolation.
- VMEbus BERR signal connection to the SCV64 controller.
- SBus/VMEbus address space allocation of SBus slots #0 and #1.

All other VMEbus interface related options are configured using extensions to the Sun OpenBoot PROM monitor program. OBP stores system configuration parameters in non-volatile storage (NVRAM) using a `setenv` mechanism familiar to UNIX shell users.

The default configuration upon delivery of the 5/64 (or after restoring the factory default values by pressing `L1-N`) is:

- VME slave accesses disabled
- VME slave base address at 0x00
- Mailbox interrupts disabled
- Mailbox interrupt at CPU level 9

This configuration is typically sufficient for simple VME configurations with a limited number of I/O boards present. For complex configurations and implementations requiring slave accesses to the 5/64 DVMA space, this default configuration may require modification. Themis has implemented the following environment variables and OBP functions for this purpose (see also *OpenBoot PROM Extension Commands on page I-141*):

- `vme32-slave-base` sets the slave base address for the 5/64 in A32 addressing mode.

- vme24-slave-base sets the slave base address for the 5/64 in A24 addressing mode.
- *cf-slot0* and *cf-slot1* are variables holding a 16 character configuration string that allocates each 16MB segment of SBus slot #0 and SBus slot #1 to either the SBus (indicated by a *'s'*) or the VMEbus (indicated by a *'v'*).
- `add-vme` prompts for the necessary information to build an entry into the VME MMU table for a specific VME board.
- `remove-vme` erases a VME boards information from the VME MMU table so it can be removed from the VME chassis.
- `show-vme` displays the contents of all VME boards configured using *add-vme* in tabular format.
- *dump-vme-map* dumps the contents of the VME MMU map table which specifies all SBus and VMEbus map specifications

The OBP command **`setenv`** must be used to set the values of the environment variables. The **`printenv`** command will list all supported environment variables and can be used to verify proper setting. You must be at the OpenBoot command prompt to enter and execute OpenBoot commands.

If autoboot is enabled, interrupt the boot sequence by pressing `L1-A` (`STOP-A`); on an serial terminal press `BREAK`.

If BOOTMON compatibility mode is enabled, you will initially see the BOOTMON prompt. Enter n to start OpenBoot:

```
Type b (boot), c (continue), or n (new command mode)
>n
ok>
```

At the ok> prompt you are now able to enter OBP commands. Use **`setenv`** to modify the environment variables necessary to configure the 5/64 for your VMEbus configurations or execute the appropriate OBP commands listed above.

The following example moves the slave window for A32 accesses to 0x80000000 and enables slave accesses:

The OBP automatically programs the SCV64 interface chip with the correct register values and retains your settings in NVRAM.

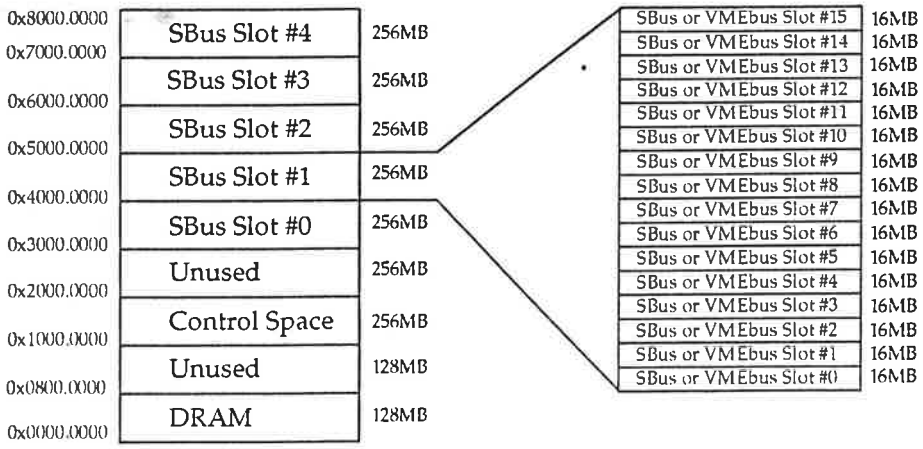
```
ok> setenv vme32-slave-base 8000.0000
    -
```

Caution – Unless you are familiar with the Forth Monitor and are experienced in interacting with your system PROM, restrict yourself to the most basic Forth Monitor operations. That is, to syncing your disks, ejecting floppies from the diskette drive, booting your system and configuring the VME interface. More advanced commands can do damage to your system's operation.

5.5.1 *VMEbus Memory Allocation*

The address space defined by the microSPARC II processor is primarily divided into on-board memory, control space and 5 SBus slots of 256MB size each. For maximum flexibility in configuring the available address map between SBus and VMEbus, each 256MB SBus address range is logically divided into 16 segments of 16MB each. Mapping of each 16MB segment to SBus or VMEbus is controlled by the VME MMU translation table (controlled through OBP commands) or hard-coded in hardware.

Figure 5-6 microSPARC II Address Space Mapping Example.



Themis has implemented a flexible VME MMU mapping table that allows users to allocate the 5 SBus slot address spaces of 256MB each, in 16MB increments to either the SBus or the VMEbus. When allocated to the VMEbus, each 16MB sector can generate any legal VME addressing mode.

SBus Address Mapping

SBus Slot #0

This slot offers full SBus/VMEbus address selection capabilities. Each 16MB segment can be individually configured to access SBus slot #0 address space or VMEbus address space.

Caution – The partitioning between SBus and VMEbus of SBus slot #0 is controlled by solder bead F3501 on the KSIC interface. If the bead is installed, slot #0 is completely mapped to the physical SBus slot.

SBus Slot #1

This slot offers full SBus/VMEbus address selection capabilities. Each 16MB segment can be individually configured to access SBus slot #1 address space or VMEbus address space.

Caution – The partitioning between SBus and VMEbus of SBus slot #1 is controlled by solder bead F3502 on the KSIC interface. If the bead is installed, slot #1 is completely mapped to the physical SBus slot.

SBus Slot #2

The entire 256MB address space is dedicated to the VMEbus. Each 16MB segment can be configured for a specific VMEbus target address and corresponding addressing mode and data width.

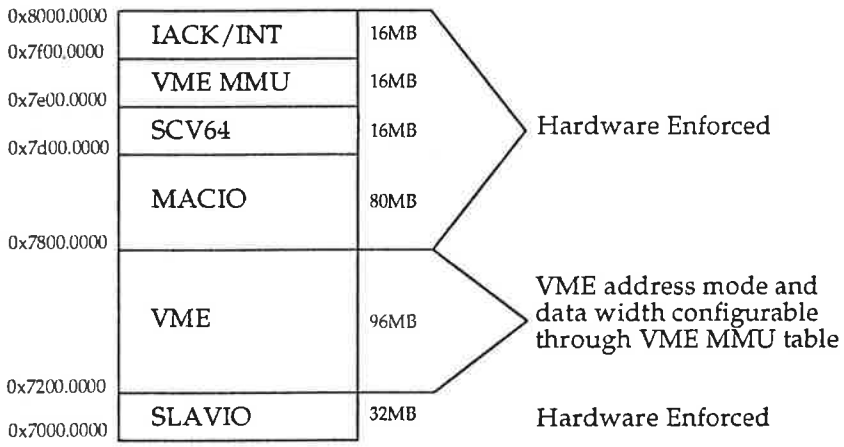
SBus Slot #3

Configuration of slot #3 is identical to SBus slot #2.

SBus slot #4

This slot is dedicated to local and VMEbus resources. The resource allocation mapping of this slot is enforced by hardware. Only the VME address target parameters can be configured in the VME MMU table.

Figure 5-7 SBus Slot #4 Resource Map



VME MMU Address Translation Table

The limited address space available to the SPARC processor consists of 5 SBus slots of 256MB each. In contrast, the VMEbus not only offers a continuous 4GB address space, but also features different address spaces that further expand its capabilities.

To facilitate flexible mapping of portions of the available SBus address space to selectable VMEbus address ranges, the 5 SBus windows have been divided into a total of 80 segments of 16MB each. Each segment is individually configurable to access the corresponding SBus slot or the VMEbus.

Mapping of memory accesses is accomplished by the VME MMU, a memory management unit maintained as a table of mapping specifiers, one for each 16MB segment.

The VME MMU table contains a 32-bit specifier entry for each 16MB segment, describing the segments mapping properties. Each specifier consists of:

- Target designation. Indicates local resources, SBus board or VME resources (unless selection is prohibited by hardware)
- Addressing mode.
- Data width. Indicates D32 or D16 mode on VMEbus accesses
- VME base address. The 16MB aligned base address generated on the VMEbus (for VME accesses only)
- Jumper configuration status. Indicates whether configuration solder beads on the KSIC for SBus slots #0 and #1 are installed.

To facilitate easy configuration of the VME MMU table, a number of OBP commands have been added to allow interactive configuration of the VME MMU table. A configured VME MMU can then be permanently stored in NVRAM.

The OBP extension commands used to configure and modify the VME MMU table are:

- `add-vme`
- `remove-vme`
- `show-vme`

For each board that is added to a VMEbus chassis, `add-vme` must be executed to install the slaves VME base address, address size and modifiers for the board being added.

Correspondingly, `remove-vme` is run when a card is removed from a VME chassis.

Please see Appendix I, *OpenBoot PROM Extension Commands on page I-141* for details on the Themis specific OpenBoot commands and their usage.

5.6 *Installing The 5/64 Paddle Board*

The paddle board delivered with the 5/64 attaches to the rear of the P2 backplane and provides industry standard connectors for serial ports C&D, Ethernet, SCSI, and printer.

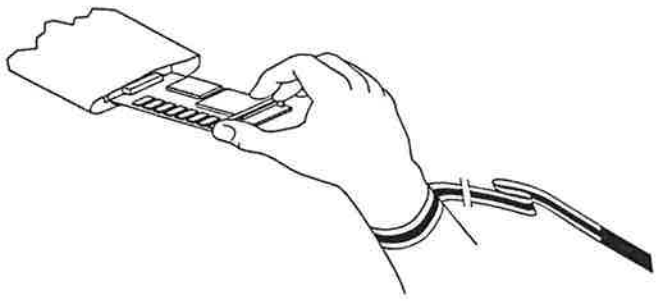
The board contains no statically sensitive components but should be handled with care to avoid bending pins on the connectors.

Two P2 connectors allow secure installation of the paddle at the rear of the P2 backplane.

5.7 *Installing An SBus Card*

Two single slot SBus cards or one dual-slot SBus card can be attached to the 5/64 via the SBus connectors[1].

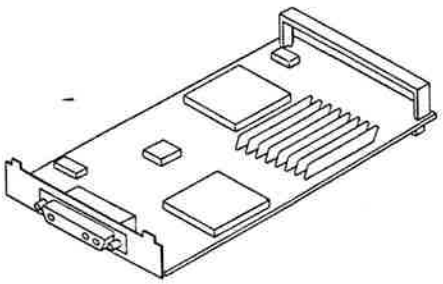
Remove the SBus card from its anti-static packaging.



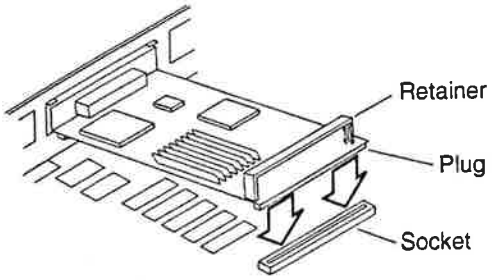
Typically an SBus card includes a metal front panel and a plastic handle on the component side to facilitate easy installation and removal of the card.

Remove the filler plate in the front panel of the 5/64 corresponding to the SBus slot(s) you wish to install your card in. Retain the mounting screws as you will need them to secure the card you are installing.

1. 5/64 version with 4 SIMM sockets support a one single-slot SBus card only

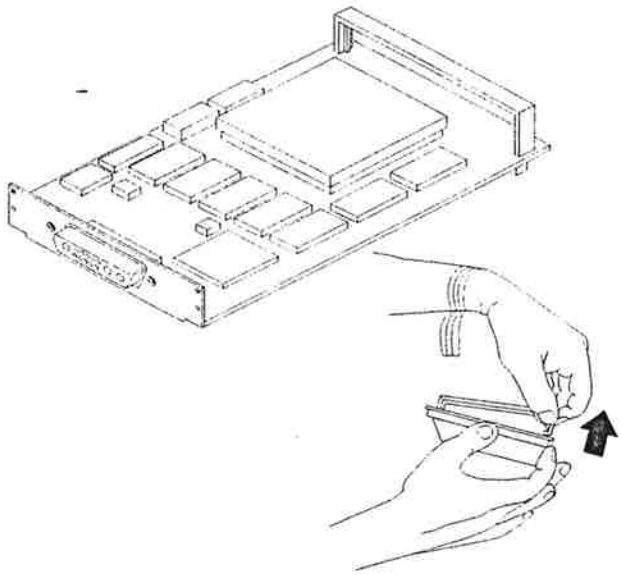


To install the SBus card, insert its front panel into the 5/64 front panel cutout and insert the SBus plug into the SBus socket on the 5/64 while holding the card by its plastic retainer. Secure the card to the front panel with the mounting screws supplied with the SBus card.



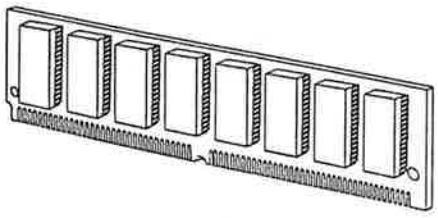
Caution – Observe all necessary anti-static measures to prevent damage to the 5/64 and the SBus card being installed.

In a full populated VME chassis, with slot space at a premium, the plastic retainer of certain SBus cards may prevent additional VME boards from properly inserting in the adjacent slot. Should this occur, simply remove the plastic retainer from the SBus card by pulling each end of the retainer from its mounting hole.



5.8 *Installing SIMM Memory*

The on-board memory capacity of the 5/64 can be expanded by installing two or four SIMM modules (4MB or 16MB per module).



The 5/64 utilizes standard 33-bit Sun SIMM modules (also available from 3rd party sources).

The DRAM address space of the microSPARC II processor is divided into 4 DRAM banks (bank0 through bank7) of 32MB each.

DRAM bank0 is on-board memory of either 8MB or 32MB size. DRAM banks 1 and 3 can be mounted on the two #1 SIMM sockets (IC301 and IC302) and bank #2 can be mounted on the two #2 SIMM sockets (IC303 and IC304).

The SIMM sockets #1 support 4MB, 16MB and 32MB of Sun compatible memory modules. The horizontal SIMM socket used on the SPARC 5/64 board is AMP horizontal SIMM connector, 7-382486-2. The approved vendors list for double, and single SIMMs is listed below:

Table 5-1 Approved list of vendors

Vendor Name	Vendor Address	Vendor Telephone #	SIMM #
CENTON Electronics	1731 Technology Drive, Suite 570, San Jose, CA 95110	(408) 441-1200	16MB SIMM Modules: CUN 16M / SPARCLX (2x 8MB CUN 8M / SPARCLX (2x 4MB) CUN 32M / SPARCLX (2x 16MB)
Kingston Technology Corp.	17600 Newhope St. Fountain Valley, CA 92708	(714) 435-2600	KTS-4000 / LX

Caution – Due to physical design limitations, installation of SIMM modules in bank #2 and/or #3 and/or #4 prevent use of an SBus card in SBus slot 0.

All possible memory configurations for the 5/64 are shown in the table below:

V = Vertical SIMM H = Horizontal SIMM

Table 5-2 Memory Configuration

	Capacity	Onboard	SIMM #1	SIMM #2	SLOT #0	SLOT #1
H	8MB	8MB	0	0	Yes	Yes
H	16MB	8MB	8MB	0	Yes	Yes
V	24MB	8MB	16MB	0	No	Yes
H	40MB	8MB	32MB	0	Yes	Yes
V	72MB	8MB	64MB	0	No	Yes
V	24MB	8MB	8MB	8MB	No	Yes
V	32MB	8MB	16MB	8MB	No	Yes
V	48MB	8MB	32MB	8MB	No	Yes
V	80MB	8MB	64MB	8MB	No	Yes
V	56MB	8MB	16MB	32MB	No	Yes
V	72MB	8MB	32MB	32MB	No	Yes
V	104MB	8MB	64MB	32MB	No	Yes
H	32MB	32MB	0	0	Yes	Yes
H	40MB	32MB	8MB	0	Yes	Yes
V	48MB	32MB	16MB	0	No	Yes
H	64MB	32MB	32MB	0	Yes	Yes
V	96MB	32MB	64MB	0	No	Yes
V	48MB	32MB	8MB	8MB	No	Yes
V	56MB	32MB	16MB	8MB	No	Yes
V	72MB	32MB	32MB	8MB	No	Yes
V	104MB	32MB	64MB	8MB	No	Yes
V	80MB	32MB	16MB	32MB	No	Yes
V	96MB	32MB	32MB	32MB	No	Yes
V	128MB	32MB	64MB	32MB	No	Yes

5.9 *Attaching the 5/64 To A Network*

The 5/64 features both a 10Base-T front panel connector as well as a AUI interface connector on the paddle board. An auto-select circuitry determines the active interface upon power-up. The default interface is 10Base-T and the AUI interface is only selected if a drop cable and media attachment unit (MAU) are connected to the paddle board or front panel AUI.

After attaching the 5/64 to the network, you can verify proper physical connection by executing the Forth monitor network selftest function:

```
ok> test net

Using AUI Ethernet Interface
Lance Register test -- succeeded
Internal loopback test -- succeeded
External loopback test -- Lost Carrier (transceiver cable problem?)
send failed

Using TP Ethernet Interface
Lance Register test -- succeeded
Internal loopback test -- succeeded
External loopback test -- succeeded
send ok
net selftest succeeded
```

This test will indicate external loopback failure on the selected network interface if no proper physical connection exists. As only one interface can be active, the unattached network interface will always return an external loopback error.

5.10 *Attaching a Keyboard and Mouse*

A standard SUN type 4 or type keyboard/mouse combination can be attached to the front panel keyboard connector.

Caution – The POST (power on self test) displays the message "`Warning: no keyboard detected`" if no keyboard is attached to the 5/64. This message may alternately indicate that configuration jumpers J3002 and J3001 are not configured for the front panel mouse/keyboard interface.

Themis SPARC 5/64 Programmers Guide 6

This guide is intended to provide helpful information and tips to systems programmers who use the Themis SPARC5/64 boards. The next section is intended to address the programmers who wish to program the Themis SPARC 5/64 board for the VMEBus without the aid of an operating system or real-time kernel. Further sections are intended to address the programmers who wish to access the Themis SPARC5/64 board from any of the supported Operating System environments.

Detailed information concerning on-board device interfaces, memory maps and device accesses are contained in the appendices and should be consulted while reading this guide.

The Themis SPARC5/64 microSPARC II processor board implements a Sun SPARCstation 5 compatible workstation on a single- or double-width 6U VMEbus board. A full VME master/slave interface is included, using the SCV64 VMEbus controller paired with the Themis KSIC SBus to SCV64 interface.

This guide is primarily concerned with the implementation and programming of 5/64 specific features that are not common to the microSPARC II family of workstations (SPARCstation 5).

6.1 5/64 VME Interface

6.1.1 Primary Features

The VME interface is composed of the SCV64 VME interface controller chip and the bus protocol converter logic responsible (KSIC) for bridging the SCV64 local bus to the SBus. For details on the SCV64 VMEbus controller please refer to *VMEBus Interface Implementation on page D-87* or the *Newbridge SCV64 Users's Manual. (NewBridge Microsystems, 603 March Road, Kanata ON K2K 2M5 Canada).*

The SCV64 offers the following major features:

- VMEbus system controller functions
- VMEbus master interface capability
- VMEbus slave interface capability
- VMEbus interrupter
- VMEbus interrupt handler
- Build-in DMA engine

Other attractive features:

- Automatic VMEbus Syscon identification
- Location Monitor
- VME slave access protection
- Auto-ID VME slot identification
- Bus isolation

6.1.2 System Controller

The 5/64 features a slot one system controller including SYSCLK, SYSRESET, bus time-out and four level arbiter. System controller capabilities are enabled by installing jumper J2001 1-2 (see *5/64 Jumper Settings on page G-123*)

Both single arbitration or round-robin arbiters are supported.

Slot one system controller features include:

- Programmable VMEbus arbitration schemes:
  - Full priority on all four request levels
  - Round-robin priority on all four request levels
  - Variant of both
- VMEbus arbitration timer (16µs):
  - 16µs-Maximum bus grant propagation time through the daisy-chain
- Programmable VMEbus request mode:
  - Unconditional (default setting)
  - Fair mode
- Programmable VMEbus ownership period:
  - No time-out
  - 2 µs
  - 4 µs
  - 8 µs (default setting)
- Programmable VMEbus data transfer time-out period:
  - 16 µs
  - 32 µs
  - 64 µs (default setting)
  - No time-out
- Programmable VMEbus release mode
  - Release when done (RWD)
  - Release on request (ROR) default setting
  - Release on Bus Clear
- System clock driver (16 MHz)
- IACK daisy chain driver
- Sys-reset generation

6.1.3 *VMEbus Master Interface*

The master interface of the 5/64 supports 8, 16 and 32-bit data transfer cycles in A32, A24 and A16 addressing modes. In DMA mode, A64 addressing capabilities are also supported.**(Note: A16/D32 transfer is not supported).**

The VMEbus is made accessible to the microSPARC II processor by mapping the 256MB address spaces of SBus slots through the VME MMU translation table map. This MMU allows the user to selectively map each 16MB segment of an SBus slot to local resources (aka SBus addresses) or to the VMEbus.

Because of the limited local address space (composed of the 5 available SBus slots of 256MB each) compared to the 4GB space of the A32 VME address space and the fact that the VMEbus supports multiple addressing modes, a flexible mapping scheme, utilizing the VME MMU mapping table concept, has been developed to allow mapping of varying portions and combinations of the 3 primary VME addressing modes (A32, A24, A16) and their sub-modes.

Themis has implemented logic to interface the SBus based addressing of the SPARC microSPARC II processor to the 68K-oriented local bus of the SCV64 VME controller. This interface, named KSIC is described in detail.

The KSIC interface and the SCV64 controller operate in conjunction with the VME MMU map table to provide flexible mapping of SBus address ranges to non-contiguous VMEbus addresses. As described earlier, each SBus slot is divided into 16MB segments. Without customer configuration, the VME MMU map is initialized from EPROM to the following factory default values:

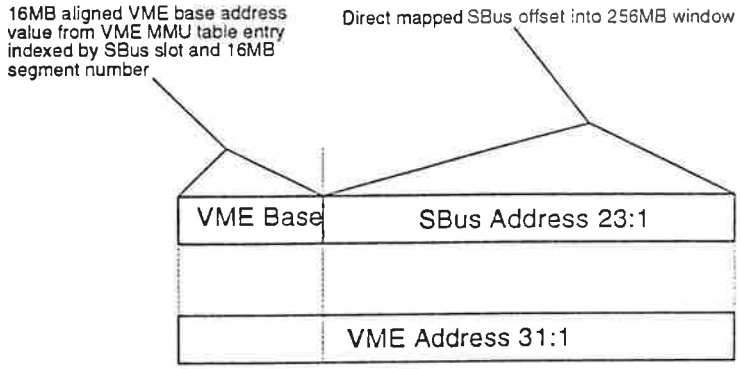
- Sbus slot #4 configured for A32:D16 on segments 2,3, and 4
- SBus slot #2 and #3 configured for A32:D16 on segments 0 through 15
- Sbus slots #0 and #1 configured for SBus access on segments 0 through 15
- VME Base Address for all A32 entries set to 0xFF
- VME Base Address for all A16:D16 entries set to 0xFF
- VME Base Address for all A24:D32 entries set to 0xF9
- VME Base Address for all A24:D16 entries set to 0xF8

When the CPU initiates a transfer to a local SBus address which is mapped to the VMEbus, the physical address bits PA24:27 as well as the select lines are used to index one 16MB segment in the VME MMU map table.

If the segment specifier entry confirms a VMEbus target address, bits 0-7 (VME Base Address) of the specifier are loaded onto the KBus address lines KA24-31. For more details on the SCV64 controller operation, please refer to *VMEBus Interface Implementation on page D-87.*

The VME address is derived as follows:

Figure 6-1 VME Master Address Generation


16MB aligned VME base address value from VME MMU table entry indexed by SBus slot and 16MB segment number
Direct mapped SBus offset into 256MB window
VME Base
SBus Address 23:1
VME Address 31:1

VME master accesses can be initiated by the CPU or the SCV64 built-in DMA engine.

The SCV64 offers the following features:

Table 6-1 Addressing Mode

Addressing mode	Source Access
A64, A32, A24	DMA
A32, A24, A16	CPU

Table 6-2 Data Transfer Mode

Data transfer mode	Source Access
D64 (MBLT), D32 (BLT),D16(BLT)	DMA
D32, D16, D08 (EO)	CPU

The following table gives the various combinations available:

EVEN / ODD

Table 6-3 Data Transfer Combinations

CPU Source Transfer		DMA Source Transfer	
A32/A24	:D32	A64/A32	:D64
A32/A24/A16	:D16	A32	:D64
A32/A24/A16	:D08 (EO)	A32/A24	:D32 (BLT)
		A32/A24	:D32
		A32/A24	:D16 (BLT)
		A32/A24	:D16

Master Block transfers are not supported by the CPU (programmed I/O). Only DMA-controlled transfers can make efficient use of VME block transfer mode.

All CPU accesses to VME are run in supervisor data mode as indicated in the following table.

Table 6-4 Master Block Transfers

Transfer Type	Address Modifier
A16	2D
A24	3D
A32	0D

DMA-sourced VME transfers may be run either in supervisor or non-privileged data mode.

Table 6-5 DMA-Sourced VME Transfers

Transfer Type	Address Modifier
A24	3D or 39
A24 BLT	3F or 3B
A32	0D or 09
A32 BLT	0F or 0B
A32 MBLT	0C or 08
A64 MBLT	00

Note – A24 MBLT mode is not supported

Addressability

For CPU-sourced transfers, the 5/64, along with the VMMU implemented in the KSIC interface, is capable of addressing from 560MB up to 1072MB in A32 mode, 16MB in A24, D16, 16MB in A24D32 and 64KB in A16 mode. The address translation mechanism is detailed later in *Translation Table on page 6-61*. Through the VME MMU, any 16MB page within the 4GB A32 VME address space becomes accessible with the exception of pages F8, F9, FF dedicated to A24D16, A24D32, A16D16 respectively.

Note – Any 32-bit access to a D16 target will cause two back-to-back transfers over the VME, taking advantage of microSPARC II dynamic bus sizing capability.

For DMA-sourced transfer, the 5/64 has no limitation; the full 4GB (A32 mode) and 16 Exabytes (A64 mode) address spaces are accessible on the VMEbus.

Caution – DMA block must be aligned on a 32-bit boundary and its byte count must be a multiple of 4-bytes.

No support for Read-modify-write transfer, as well as, unaligned transfer is provided.

6.1.4 *VME MMU Address Translation Table*

The limited address space available to the SPARC processor consists of 5 SBus slots of 256MB each. In contrast, the VMEbus not only offers a continuous 4GB address space, but also features different address spaces that further expand its capabilities.

To facilitate flexible mapping of portions of the available SBus address space to selectable VMEbus address ranges, the 5 SBus windows have been divided into a total of 80 segments of 16MB each. Each segment is individually configurable to access the corresponding SBus slot or the VMEbus.

Mapping of memory accesses is accomplished by the VME MMU, a memory management unit maintained as a table of mapping specifiers, one for each 16MB segment. The VME MMU mapping table is located at 0x7E00.0000 (see SBus slot #4 mapping below). The address for the specifier of a specific SBus

slot number and block segment offset can be computed as a function of the slot number (integer range 0 through 4) and segment number (integer range 0 through 15):

Specifier Address = f(SBus slot #, block #) = f(s,w)

This translates to the following formula for computing the address of a segment specifier using integer arithmetic:

addr = 0x7E00.0000 + ((31 - 2^s) ¥ 32) + (4 ¥ ($w^3$2)) + ((w % 2) ¥ 0x400)

The following table gives the addresses of each specifier versus the slot number and block segment offset.

Figure 6-2 Default VME MMU Table

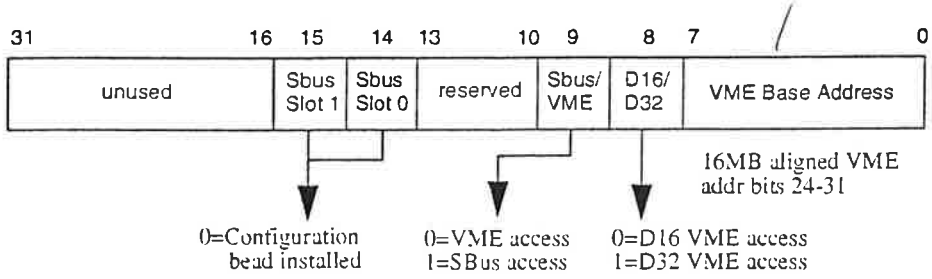
Addr	Slot	Block	Addr	Slot	Block
0x7E00.03C0	#0	#0	0x7E00.0370	#2	#8
0x7E00.07C0	#0	#1	0x7E00.0770	#2	#9
0x7E00.3C4	#0	#2	0x7E00.0374	#2	#10
0x7E00.07C4	#0	#3	0x7E00.0774	#2	#11
0x7E00.03C8	#0	#4	0x7E00.0378	#2	#12
0x7E00.07C8	#0	#5	0x7E00.0778	#2	#13
0x7E00.03CC	#0	#6	0x7E00.037C	#2	#14
0x7E00.07CC	#0	#7	0x7E00.077C	#2	#15
0x7E00.03D0	#0	#8	0x7E00.02E0	#3	#0
0x7E00.07D0	#0	#9	0x7E00.06E0	#3	#1
0x7E00.03D4	#0	#10	0x7E00.02E4	#3	#2
0x7E00.07D4	#0	#11	0x7E00.06E4	#3	#3
0x7E00.03D8	#0	#12	0x7E00.02E8	#3	#4
0x7E00.07D8	#0	#13	0x7E00.06E8	#3	#5
0x7E00.03DC	#0	#14	0x7E00.02EC	#3	#6
0x7E00.07DC	#0	#15	0x7E00.06EC	#3	#7
0x7E00.03A0	#1	#0	0x7E00.02F0	#3	#8
0x7E00.07A0	#1	#1	0x7E00.06F0	#3	#9
0x7E00.03A4	#1	#2	0x7E00.02F4	#3	#10
0x7E00.07A4	#1	#3	0x7E00.06F4	#3	#11
0x7E00.03A8	#1	#4	0x7E00.02F8	#3	#12
0x7E00.07A8	#1	#5	0x7E00.06F8	#3	#13

Addr	Slot	Block	Addr	Slot	Block
0x7E00.03AC	#1	#6	0x7E00.02FC	#3	#14
0x7E00.07AC	#1	#7	0x7E00.06FC	#3	#15
0x7E00.03B0	#1	#8	0x7E00.01E0	#4	#0
0x7E00.07B0	#1	#9	0x7E00.05E0	#4	#1
0x7E00.03B4	#1	#10	0x7E00.01E4	#4	#2
0x7E00.07B4	#1	#11	0x7E00.05E4	#4	#3
0x7E00.0388	#1	#12	0x7E00.01E8	#4	#4
0x7E00.0788	#1	#13	0x7E00.05E8	#4	#5
0x7E00.03BC	#1	#14	0x7E00.01EC	#4	#6
0x7E00.07BC	#1	#15	0x7E00.05EC	#4	#7
0x7E00.0360	#2	#0	0x7E00.01F0	#4	#8
0x7E00.0760	#2	#1	0x7E00.05F0	#4	#9
0x7E00.0364	#2	#2	0x7E00.01F4	#4	#10
0x7E00.0764	#2	#3	0x7E00.05F4	#4	#11
0x7E00.0368	#2	#4	0x7E00.01F8	#4	#12
0x7E00.0068	#2	#5	0x7E00.05F8	#4	#13
0x7E00.036C	#2	#6	0x7E00.01FC	#4	#14
0x7E00.076C	#2	#7	0x7E00.05FC	#4	#15

The VME MMU table contains a 32-bit specifier entry for each 16MB segment, describing the segments mapping properties.

VME BOARDS

Figure 6-3 VME MMU Map Specifier Entry



BIT #10 - SET TO SHORTEN AS TIME

Each specifier consists of:

- Target designation. Indicates local resources, SBus board or VME resources (unless selection is prohibited by hardware)
- Data width. Indicates D32 or D16 mode on VMEbus accesses
- VME base address. The 16MB aligned base address generated on the VMEbus (for VME accesses only). This field is used by the SCV64 to propagate A16, A24, A32 access to the VME. If the value is not F8, F9, FF, then an A32 transfer will be initiated. If the value is F8 (resp. F9), an A24:D16 (resp. A24:D32) transfer will be initiated. If the value is FF and SBus address bits SBA27-20 are set to 1, the SCV64 will initiate an A16:D16.
- Solder beads F3501 and F3502 configuration status. Indicates whether configuration solder beads for SBus slots #0 and #1 are installed.

Figure 6-4 VME MMU 16MB Segment Factory Defaults

Segment	SBus #0	SBus #1	SBus #2	SBus #3	SBus #4
0	A32 or SBUS	A32 or SBUS	A32	A32	SLAVIO
1	A32 or SBUS	A32 or SBUS	A32	A32	SLAVIO
2	A32 or SBUS	A32 or SBUS	A32	A32	A32
3	A32 or SBUS	A32 or SBUS	A32	A32	A32
4	A32 or SBUS	A32 or SBUS	A32	A32	A32
5	A32 or SBUS	A32 or SBUS	A32	A32	A24:D32
6	A32 or SBUS	A32 or SBUS	A32	A32	A24:D16
7	A32 or SBUS	A32 or SBUS	A32	A32	A16:D16
8	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
9	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
10	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
11	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
12	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
13	A32 or SBUS	A32 or SBUS	A32	A32	SCV64

Segment	SBus #0	SBus #1	SBus #2	SBus #3	SBus #4
14	A32 or SBUS	A32 or SBUS	A32	A32	VMEMMU
15	A32 or SBUS	A32 or SBUS	A32	A32	IACK

To facilitate easy configuration of the VME MMU table, a number of OBP commands have been added to allow interactive configuration of the VME MMU. A configured VME MMU can then be permanently stored in NVRAM.

The OBP extension commands add-vme, remove-vme,and show-vme are used to configure and modify the VME MMU table.

For each board that is added to a VMEbus chassis, add-vme must be executed to install the slaves VME base address, address size and modifiers.

Correspondingly, remove-vme is run when a card is removed from a VME chassis.

Caution – To avoid possible problems due to an uninitialized VME MMU map table, the memory management translation table is disabled and a dummy write to byte location 0x7f00.0000 is required to toggle the translation mechanism between disable/enable mode.

6.1.5 *VMEbus Slave Interface*

The VME slave interface implemented on the Themis 5/64 supports A64, A32 and A24 slave accesses. Block mode transfer capability is also supported. When transferring data using BLT mode on VME, the start address of the whle block of transferred data shall be aligned on a 8 long word boundary, and the block size shall be a multiple of 8 long words.

VME slave accesses generate local Sbus DVMA accesses. The slave base addresses and sizes of both the A32 and A24 address spaces are set through the VMEBAR register of the SCV64 VMEbus controller.

The slave addressing window is limited by hardware to either 8MB, 32MB, 64MB or 128MB selectable through setting solder beads F2501, F2502 located on the KSIC interface.

Table 6-6 Slave Window Size Configuration

F2502	F2501	Slave Window Size
OFF -	OFF	8 MB (default setting)
OFF	ON	32 MB
ON	OFF	64 MB
ON	ON	128 MB

The SCV64 offers the following features in support of slave accesses:

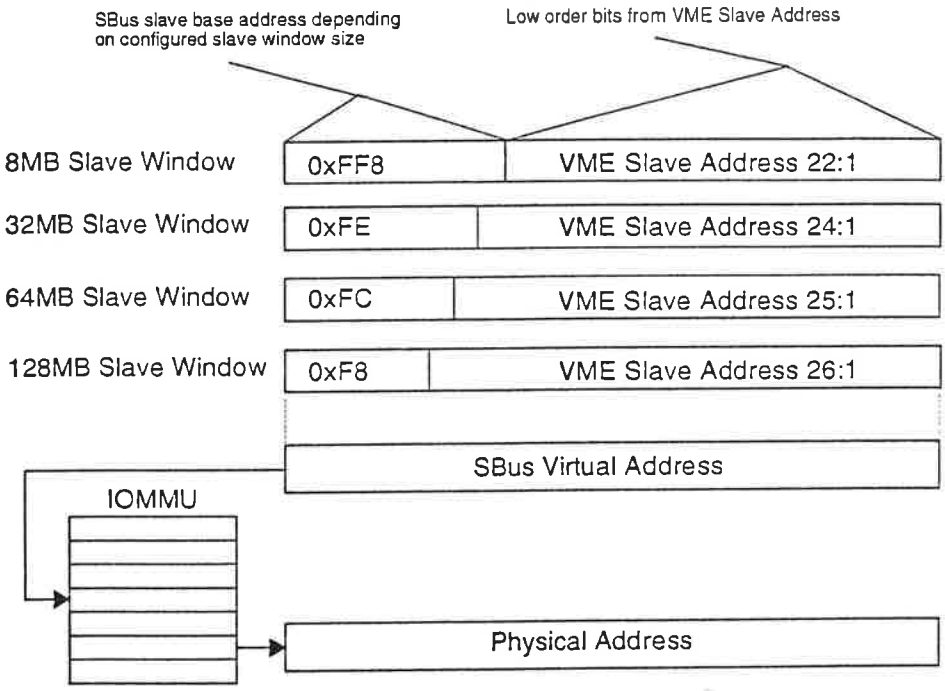
- VME slave access protection
- Programmable VME slave access window size
- Addressing mode A64, A32, and A24 (no A16)
- Support BLT transfer
- Location monitor with 32-deep message FIFO for inter-process communication

Slave address programming information are summarized in the *SCV64 user manual*. A sophisticated protection scheme could be implemented using the SCV64's APBR register. It is not supported by Solaris or VxWorks.

Every slave VME access is first validated by the SVC64. It will ignore any access outside the programmed slave VME access windows; therefore the access may time-out on the VMEbus. The SCV64 will return a bus error without initiating a local transaction if the access is protected by the APBR register or if its location monitor FIFO is full. In all other cases, it will propagate the VME slave access to the SBus through the KSIC interface and become a SBus master running a DVMA access (Direct Virtual Memory Access).

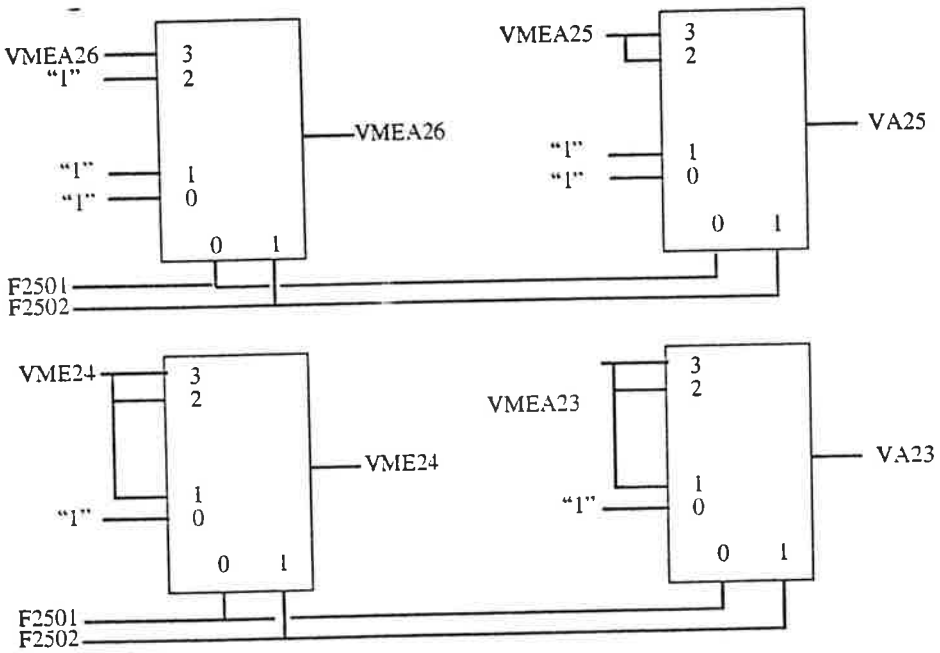
The virtual SBus address is assembled from the low order bits of the VMEbus address and the local slave window base address which is dependant on the configuration of the F2501 and F2502 solder beads. The resulting virtual SBus address is finally translated into a physical address by the IOMMU.

Figure 6-5 VME Slave Address to Sbus Mapping



The mechanism to convert a slave VME address into a SBus virtual address relies on the setting of the solder beads F2501 and F2502 on the KSIC.

Figure 6-6 Slave Address Decoding



Under Solaris 1.1, only a 7MB slave window size is supported. The window is divided into A24 and A32 slave access:

Table 6-7 Solaris 1.1 Slave Accesses

Transfer Type	Slave Window Address Range
A24	FF80.2000 to FF8F.FFFF (1MB - 8KB)
A32	FF90.0000 to FFEF.FFFF

Under Solaris 2.4, only an 8MB slave window size is supported. The window is divided into A24 and A32 slave access:

Table 6-8 Solaris 2.x Slave Accesses

Transfer Type	Slave Window Address Range
A24	FF80.0000 to FF8F.FFFF
A32	FF80.0000 to FFFF.FFFF

The IOMMU translation table must be configured for the virtual DVMA address generated by a VME slave access. If there is no corresponding entry in the IOMMU, the VME slave access transfer will cause generation of a bus error by the SBus controller.

Slave RMW transfers are not supported.

Use of the OBP variables vme32-slave-base, and vme24-slave-base (described in *OpenBoot PROM Extension Commands on page I-141*) allows transparent setting of the correct VMEBAR values of the SCV64.

Caution – Any attempt to access to its own slave VME access window results in a bus error on the SBus.

6.1.6 *VMEbus/SCV64 Interrupts*

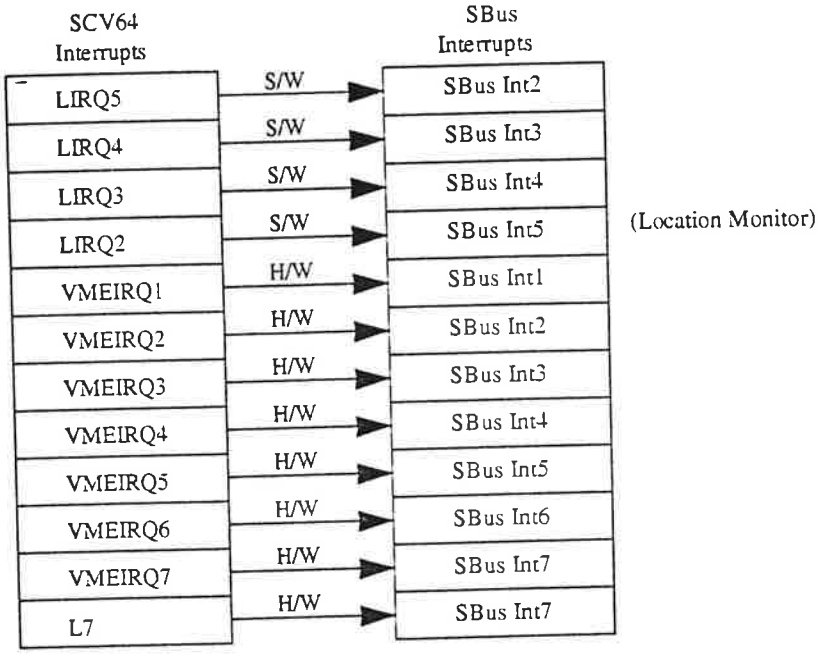
The VMEbus interface on the 5/64 supports all interrupt levels implemented by the VMEbus. VMEbus interrupts are converted to their corresponding SBus interrupt levels as described below. The status of each SCV64 generated interrupt is shown in the interrupt status register. After detecting a VMEbus interrupt, it is typically acknowledged by accessing the interrupt acknowledge register corresponding to the interrupt.

The SCV64 VMEbus controller can accept interrupt requests from three groups of interrupt sources:

- Level 7 interrupt (Abort Switch, ACFAIL, BIMODE, SYSFAIL)
- General local interrupts (LIRQ0 - LIRQ5) only LIRQ 2, 3, 4, 5 available
- VMEbus interrupts (VMEIRQ1 - VMEIRQ7)

SCV64 interrupts are mapped to their corresponding SBus interrupt levels as follows:

Table 6-9 SCV64 to SBus Interrupt Levels



"H/W" means the argument is forced by hardware.

"S/W" means the argument is forced by software.

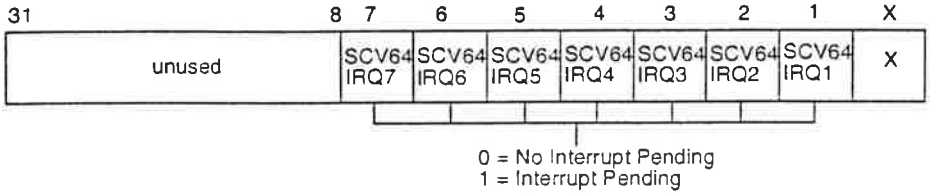
The general local interrupts are allocated as:

- LIRQ5 is connected to the Watchdog output
- LIRQ4 is connected to the tick timer output
- LIRQ3 is connected to the VMEINT output. This output is active whenever a VMEbus related event (i.e. DMA finished, bus error, etc.) has occurred
- LIRQ2/KIACK is configured as KIACK. The LIRQ2 interrupt is not available externally. It is associated internally to the mailbox interrupt..
- LIRQ1 and LIRQ0 are not used and pulled to Vcc.

VME Interrupt Status

An 32-bit register is provided to read the status of the VMEbus interrupt request lines IRQ1 through IRQ7. The VME Interrupt Status register is located at 0x7F0F.0400 and defined as follows:

Figure 6-7 VME Interrupt Status Register



VMEbus Interrupt Acknowledgment

The 5/64 implements a total of 14 interrupt acknowledge registers. Seven of these registers are dedicated to local interrupt sources (i.e., DMA, location monitor, etc.). The other seven registers are assigned to VME interrupt acknowledgment.

The base address of the VME interrupt acknowledge registers is 0x7F0F.0000 and VMEbus interrupt level n generated by the SCV64 is acknowledge by the CPU reading a 32-bit (resp. 8-bit) value from the location computed by:

register address = 0x7F0F.0000 (resp. 0x7F0F.0003)+ 4 * n

where n is a value between 1 and 7.

Caution – If an attempt is made to acknowledge an interrupt that is not pending, a bus error is returned.

The base address of the local interrupt acknowledge registers is 0x7F0F.0400.

Local interrupts are acknowledged by first clearing the source of the local interrupt in the SCV64 followed by reading the corresponding interrupt acknowledge register computing by:

register address = 0x7F0F.0400 + 4 * n (32-bit access) or 0x7F0F.0403 + 4 * n (8-bit access)

6.1.7 *SCV64 Inter-Processor Communication/Location Monitor*

A location monitor resides at the top longword (32bits) of each A24 and A32 slave access window and is accessible from both the VME bus and local bus.

The monitor location is implemented by a Location Monitor FIFO which consists of 31 entries of 32 bits and allow the exchange of 16-bit or 32-bit messages between processes. A write access to the location monitor enters the 32-bit value into the Location Monitor FIFO. If a 16-bit word transfer was used to write the location monitor, the upper 16-bit word is set to all one.

A read access to the location monitor address will result in a bus error.

While the Location Monitor FIFO contains entries, the SCV64 asserts the LMHD bit in the DCSR register and generates an interrupt to the local processor by asserting LMINT. LMINT is only negated at the end of a LMFIFO read when the FIFO is empty. LMINT may generate an SBus interrupt level 5.

A write attempt will time out if the FIFO is full and no entry becomes available before the time-out counter expires.

Messages are retrieved by reading from the LMFIFO register. Attempts to read from an empty FIFO will result in undefined data being returned

Caution – If the LMFIFO is read while it is empty and a message arrives from the VMEbus, then unstable data may be put on the local bus. Do not read the LMFIFO unless the LMHD bit is set and LMINT is asserted.

The local CPU may write to its own location monitor through its slave address window on the VMEbus. The SCV64 address decoder responds by redirecting the message to the LMFIFO.

Note – Mailbox registers are not supported under Solaris 1.1.1B or 2.X for compatibility with existing Sun VME interface implementations.

6.2 *Overview of VME Interface Under Solaris 1.1.1B/2.X*

Great effort has been expended to achieve complete compatibility between Sun VME implementations and the device driver interface implemented for the 5/64 board under Solaris.

This section is primarly concerned with the implementation and programming of 5/64 specific features taht are not common to the microSPARC II family of workstations (SPARCstation 5). It assumes that the reader is familiar with the sun4m architecture and the concepts of VMEbus and device drivers. A separate tutorial provides more detailed information for readers who are not familiar with these concepts.

The VMEbus interface of the 5/64 is transparently implemented under Solaris 1.X (SunOS 4.1.3) and Solaris 2.X. The software interface is fully compatible with generic Sun4m VME architecture systems. Any VME device driver that confirms to the appropriate Solaris Device Driver Interface will run unmodified on Themis platforms.

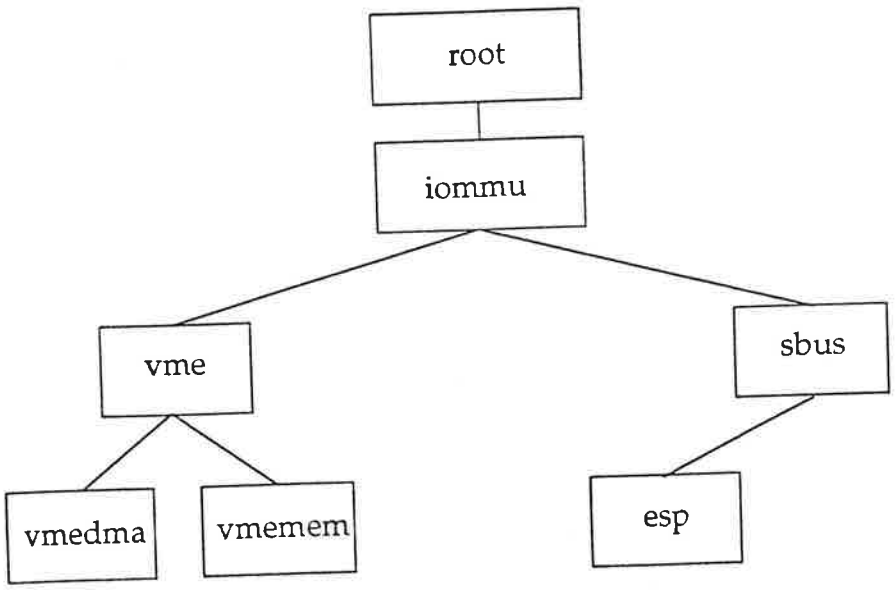
Themis Computer has developed a number of device drivers to provide the necessary software interface needed by system programmers. The drivers can be classified into three categories:

- VMEbus Nexus drivers
- Utility drivers
- Sample drivers

The drivers are described in the following subsections. Themis also provides online manual pages for all the drivers included in the software interface. These manual pages include more detailed information on the drivers.

6.2.1 *VMEbus Nexus Driver*

The VME driver forms the core of the software interface provided by Themis Computer. VME is a bus nexus driver that encapsulates all the architectural features of supporting the Themis 5/64 board on Solaris systems. This driver is configured to be permanently loaded as the bus nexus driver for the VMEbus.



Any driver that specifies its class or parent as vme will be attached to the vme driver. The vme driver provides support for all VMEbus operations, including VME interrupt processing, Direct Virtual Memory Access (DVMA) on the VMEbus, mapping of registers on the VMEbus to kernel space, mapping of VMEbus memory to user address space through mmap(), etc.

The vme driver recongnizes the standard configuration file for VME device drivers. (see vme(4)). The configuration of any vme device driver specifies the VMEBus interrupt level and the interrupt vector. The mapping of the VMEbus interrupt level to the SPARC interrupt priority level is as follows:

Table 6-10 VMEbus Interrupt Mapping

VMEBus		1	2	3	4	5	6	7
SPARC ip	1	2	3	5	7	9	11	13

6.2.2 *Utility Drivers*

These drivers are a standard part of the software interface provided by Themis Computer. These drivers provide a useful interface to system programmers who wish to utilize the different features of the VMEbus architecture.

The VME device driver provides the standard /dev devices for accessing the VMEbus from user processes:

Table 6-11 VME Device Drivers

Device File	Address Size	Data Transfer Size
/dev/vme32d32	32	32
/dev/vme32d16	32	16
/dev/vme24d32	24	32
/dev/vme24d16	24	16
/dev/vme16d32	16	32
/dev/vme16d16	16	16

Read(), write() and mmap() calls are supported and no special ioctl() calls are required to configure the interface. VME interface configuration is performed under OBP and should not be modified while the operating system is running.

The SCV driver provides access to features of the SCV64 chip, including the Direct Memory Access engine. The SCV driver is configured as a chi8ld of the VME (7) driver, and will function only on Themis platforms that feature the Newbridge SCV64 VMEbus Controller. The driver supports 16-byte aligned DMA transfers between the main memory and the VMEbus. The user can specify the direction of the transfer and the VME address space that is involved in the transfer.

6.2.3 *Sample Drivers*

The software interface provided for the 5/64 platform fully supports standard VMEbus device drivers written for Solaris environments. It is the intention of Themis Computer to fully support users who wish to write their own VME device drivers that would function on Themis SPARC 5/64 platforms. To aid these users and to illustrate the specific features of the VMEbus architecture, Them provides a number of sample device drivers. Themis provides the complete source code and the necessary configuration files for these drivers.

- vmeintl: is a sample driver provided by Themis Computer to illustrate the vectored interrupt mechanism of the VMEbus. The vmeirq driver relies on the driver configuration files to specify the interrupts it should handle. Each instance of the driver registers the interrupts with the system. Through the ioctls implemented by the driver, the user can generate interrupts and send them to the appropriate driver instances. The interrupt generator and the interrupt receiver need to run on different computers.

- vmedvma: is a sample driver provided by Themis Computer to illustrate the use of Direct Virtual Memory Access within a device driver. The user can ask the driver to allocate a DMA region on the VMEbus. The driver allocates all the necessary resources to support a DVMA transfer to this region. The driver also implements mechanisms by which the user program can writer to or read from the DVMA region.

6.2.4 *OpenBoot PROM Echancements*

The OpenBoot PROM has been enhanced with a set of environment variables that configure slave and master access windows and maintain these values in NVRAM. The kernel uses these OBP variables to initialize the VME interface. Address maps and base address are typically configured and finalized during system integration and after all VMEbus board base addresses have been configured. Run-time mapping and address changes are not required. Please see *OpenBoot PROM Extension Commands on page I-141* for details on the available OBP configuration commands. OBP retains your configuration in NVRAM. Future releases of on-line documentation regarding application and device driver programming specific.

Themis SPARC 5/64 Board Drawings

This appendix contains the assembly drawings for the Themis SPARC 5/64 board.

KSIC Top

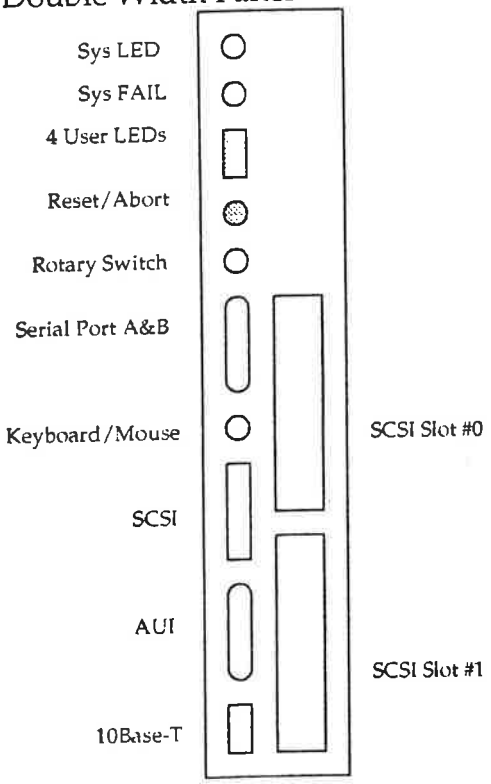
KSIC Bottom

SPARC 5/64 Top

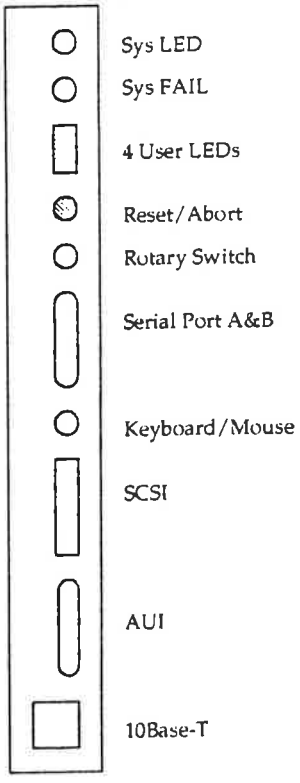
SPARC 5/64 Bottom

Themis SPARC 5/64 Front Panels

Double Width Panel



Single Width Panel



VME Access Speed

VMEbus transfer speeds measured with an 5/64 in slave mode and no SBus modules installed using a digital storagae oscillosacope.

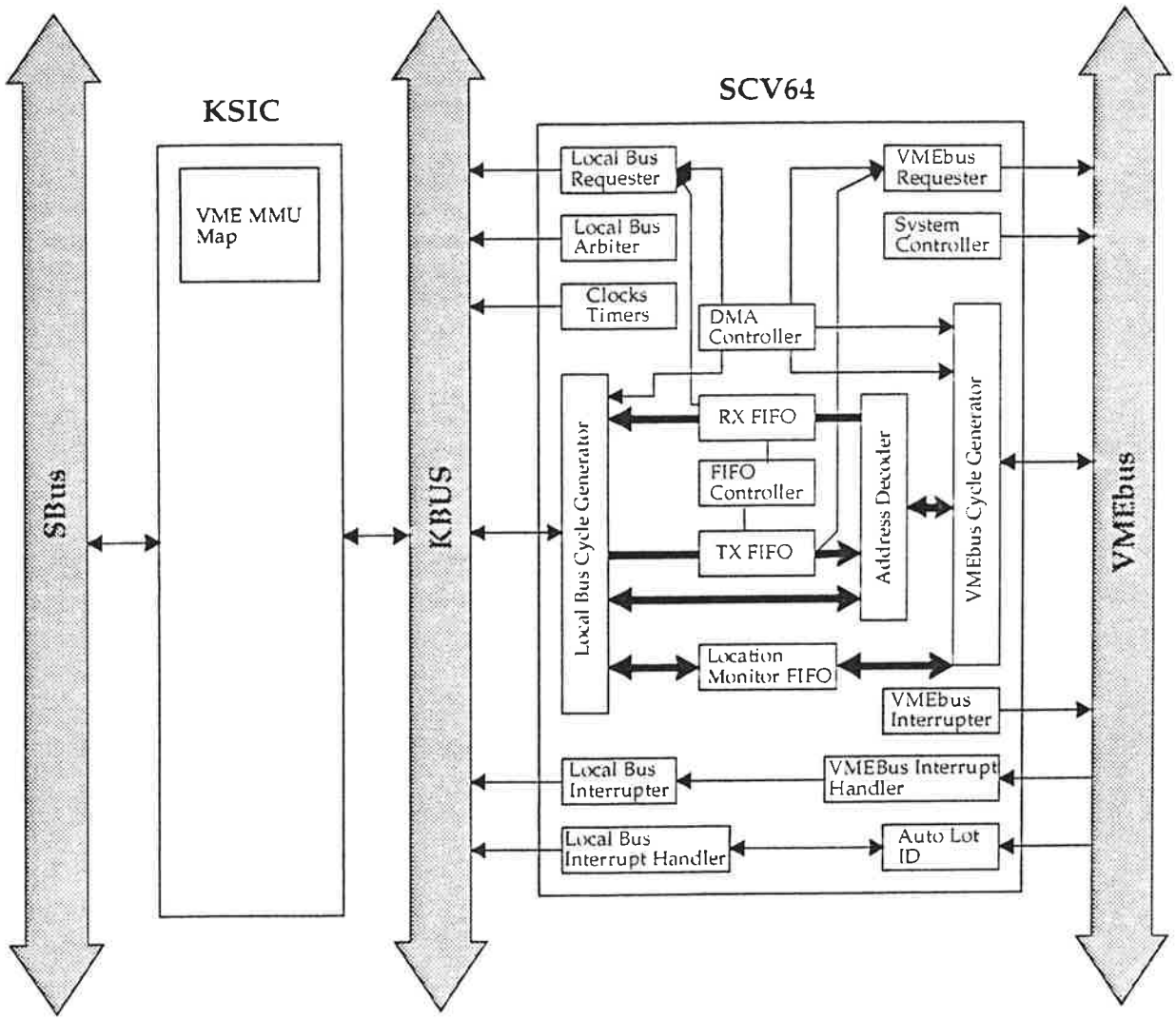
Transfer Type	Peak Rate MBytes/s	Burst Rate MBytes/s	Average Rate MBytes/s
Master VME Write (No BLT)	20		10
Master VME Read (No BLT)	4 (ROR)		4
VME⇒LXE DMA multiplexed Block Transfer	36	30	14
LXE⇒VME DMA multiplexed Block Transfer	49	38	19
Slave VME Write (no BLT)	8.8		7
Slave VME Read (no BLT)	2.7		2.6

VMEBus Interface Implementation

The appendix contains an abbreviated version of the specifications for the Newbridge SCV64 VME Controller Chip as they apply to the 5/64 design. The VMEbus implementation on the 5/64 is composed of both the SCV64 controller and the KSIC interface. Significant features involving VMEbus and SBus mapping and addressing are implemented in the KSIC interface and affect the operation and functionality of the SCV64 controller. Please refer to the programming and installation section of this manual for details on the KSIC interface.

D.1 Features

- VMEbus Rev. D compliant
- A32,-A24 D32, D16, D08 (EO) Master capabilities and A64 capability when using DMA
- A16/D16, D08 master capabilities
- A32, A24 Slave capabilties with configurable slave window sizes (8/32/64/128MB) generating DVMA accesses
- Full VMEbus system controller functionality
- Automatic VMEbus Syscon identification
- Interrupter and interrupt handling for all 7 VMEbus interrupt levels
- VME slave access protection
- Multiple VMEbus request and release options
- Multiple VMEbus arbitration schemes
- Integrated DMA controller


SBus
KSIC
VME MMU Map
KBUS
SCV64
Local Bus Requester
Local Bus Arbiter
Clocks Timers
DMA Controller
RX FIFO
FIFO Controller
TX FIFO
Local Bus Cycle Generator
Address Decoder
VMEbus Cycle Generator
Location Monitor FIFO
VMEbus Requester
System Controller
VMEbus Interrupter
VMEBus Interrupt Handler
Local Bus Interrupter
Local Bus Interrupt Handler
Auto Lot ID
VMEbus

D.2 General Description

The SCV64 is a single chip implementation of the 64-bit VMEbus interface compliant with IEEE Rev. D.

The controller encompasses a rich set of programmable features and operational modes that make it ideally suited to CPU board implementations with both master and slave capabilities.

Caution – Not all features of the SCV64 controller are applicable to the 5/64 design and certain capabilites and elements of the SCV64 have been superceded by custom implementations on the 5/64. Please consult the *Themis SPARC 5/64 Programmers Guide on page 6-55* prior to programming the SCV64.

D.3 SCV64 Registers

Table D-1 Following is a list of the SCV64 Registers:

Register	Address	Name	Register	Address	Name
DMA Local Address	0x7d00.0000	DMALAR	VMEbus Interrupter	0x7d00.008C	VINT
DMA VMEbus Address	0x7d00.0004	DMAVAR	VMEbus Requester	0x7d00.0090	VREQ
DMA Transfer Count	0x7d00.0008	DMATC	VMEbus Arbiter	0x7d00.0094	VARB
Control and Status	0x7d00.000C	DCSR	ID Register	0x7d00.0098	ID
VMEbus Slave Base Address	0x7d00.0010	VMEBAR	Control and Status	0x7d00.009C	CTL2
Rx FIFO Data	0x7d00.0014	RXDATA	Level 7 Interrupt Status	0x7d00.00A0	7IS
Rx FIFO Address Register	0x7d00.0018	RXADDR	Local Interrupt Status	0x7d00.00A4	LIS
Rx FIFOI Control Register	0x7d00.001C	RXCTL	Level 7 Interrupt Enable	0x7d00.00A8	7IE
VMEbus/VSB Bus Select	0x7d00.0020	BUSSEL	Local Interrupt Enable	0x7d00.00AC	LIE
VMEbus Interrupter Vector	0x7d00.0024	IVECT	VMEbus Interrupt Enable	0x7d00.00B0	VIE
Access Protect Boundary	0x7d00.0028	APBR	Local Interrupts 1 and 0	0x7d00.00B4	IC10
Tx FIFO Data Output Latch	0x7d00.002C	TXDATA	Local Interrupts 3 and 2	0x7d00.00B8	IC32
Tx FIFO Addr Ouput Latch	0x7d00.0030	TXADDR	Local Interrupts 5 and 4	0x7d00.00BC	IC54
Tx FIFO AM/Control Bit Latch	0x7d00.0034	TXCTL	Miscellaneous Control	0x7d00.00C0	MISC
Location Monitor FIFO Read	0x7d00.0038	LMFIFO	Delay Line Control Register	0x7d00.00C4	DLCT
SCV64 Mode Control	0x7d00.003C	MODE	Delay Line Status Register 1	0x7d00.00C8	DLST1
Slave A64 Base Address	0x7d00.0040	SA64BAR	Delay Line Status Register 2	0x7d00.00CC	DLST2
Master A64 Base Address	0x7d00.0044	MA64BAR	Delay Line Status Register 3	0x7d00.00D0	DLST3
Local Address Generator	0x7d00.0048	LAG	Mailbox Register 0	0x7d00.00D4	MBOX0
DMA VMEbus Transfer Count	0x7d00.004C	DMAVTC	Mailbox Register 1	0x7d00.00D8	MBOX1
Status Register 0	0x7d00.0080	STAT0	Mailbox Register 2	0x7d00.00DC	MBOX2
Status Register 1	0x7d00.0084	STAT1	Mailbox Register 3	0x7d00.00E0	MBOX3
General Control Register	0x7d00.0088	GENCTL			

D.4 VME System Controller

D.4.1 Bus Arbitration

The arbiter is enabled whenever the SCV64 is configured as system controller. The arbitration module provides four programmable arbitration modes.

Arbitration Modes

The SCV64 provides all arbitration modes defined in the VMEbus specification:

- Full priority
- Round Robin
- Single level arbiter

Additionally, two mixed priority modes are provided:

- Priority level 3
- Priority level 2 and 3

All arbitration modes are determined by the ARB1 and ARB0 bits in the VARB register. Since the arbiter logic continously samples the VARB register, any changes to these bits will immediately be reflected in the current arbitration mode. The default mode after reset is full Round Robin.

Table D-2 Arbitration Mode Settings

Arbitration Mode	ARB1, ARB0 Setting
Full Round Robin	00
Priority 3, Round Robin 2,1,0	01
Priority 3 2; Round Robin 1,0	10
Full Priority	11

Arbitration Time-out

Arbitration time-out ensures that bus arbitration will resume if BBSY* is not asserted within 16µs of a bus grant signal. The time-out returns the arbiter to its previous mode and outstanding requests. Arbitration time-out is enabled by default and is deactivated by clearing the ATEN bit in the VARB register.

D.4.2 Bus Timer

The SCV64 uses BERR* to terminate VMEbus data transfers if a slave does not respond within a specific programmed time. The time-out period before BERR* is driven low can be 16 µs, 32 µs, 64 µs (the default setting), or Never, depending upon the value of the VXL1 and VXL0 bits inthe VARB register. The Never setting should only be used during system debugging, since the VMEbus cannot recover from errors (such as incorrect addressing) in this mode. Note also that the Auto-ID function depends on bus time-out.

The timer is held reset while DS1* and DS0* are high, and begins running when either data strobe becomes asserted. At time-out,the SCV64 asserts and maintains BERR* until both DS1* and DS0* are high. Note that a race condition can occur if a slave takes a long time to respond and asserts DTACK* coincidentally withthe SCV64 assertion of BERR*. Under these conditions, the master may see DTACK* and BERR* simultaneously. On-board logic may be required to resolve this conflict.

Table D-3 VMEbus Time-out Setting

Time-out Period	VXL1, VXL0 Setting
Never	00
16µs	01
32µs	10
64µs (default)	11

D.5 VME Slave Interface

The VME slave image memory map may be accessed from the VMEbus as either A32 or A24. Addressing mode A16 is not supported in slave mode. The base address for the A32 and A24 slave images (also referred to as slave windows) are programmed using the VMEBAR register.

VMEBAR Register (0x7d00.0010)

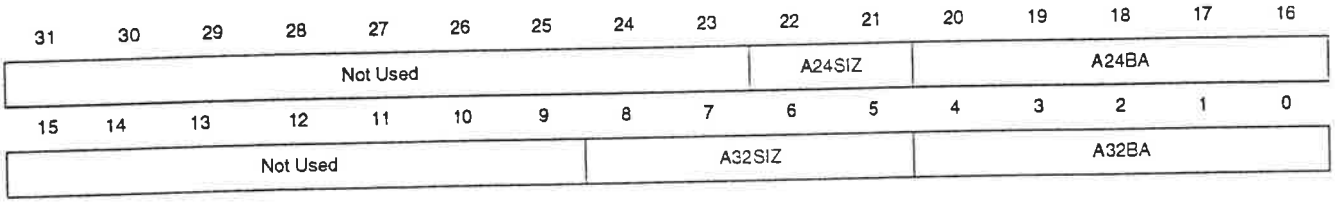


Table D-4 VMEBAR Register Settings

Name	State	Function
A24SIZ	0	512KB slave image
	1	1MB slave image
	2	2MB slave image
	3	4MB slave image
A24BA	00-1F	Sets base address of A24 slave image. A24BA will be compared against VME A23:A19
A32SIZ	xx	Sets size of A32 slave image
A32BA	xx	Sets base address of A32 slave image

The A32 image base address can be programmed to any 128MB boundary and to any size from 4KB to 128MB.

The A24 image base address can be programmed to any 512KB boundary or multiple of its programmed size, whichever is larger, within the 16MB A24 addressing space.

Caution – The KSIC interface of the 5/64 design limits the slave address window to 8MB, 32MB, 64MB or 128MB, configurable through solder beads. .

Please refer to *VMEbus Slave Interface on page 6-65* for details on the slave addressing implementation and programming of the 5/64 slave image capability.

D.6 VME Interrupter

The SCV64 can generate VMEbus interrupts on any of the seven VME interrupt levels. Interrupt level and interrupt generation are controlled by the VINT register.

VINT Register (0x7d00.008c)

7	6	5	4	3	2	1	0
0	0	0	0	INT	IL2	IL1	IL0

Table D-5 VINT Register Settings

Name	State	Function
INT	0 (read)	No interrupt pending
	1 (read)	Interrupt is presently asserted
	0 (write)	No effect
	1 (write)	Assert interrupt
IL2-0	0	Interrupter clears itself
	1-7	Sets corresponding interrupt level

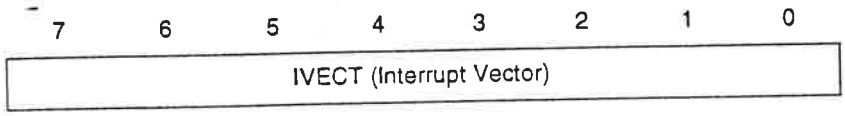
The interrupt level is encoded in bits IL2-IL0 of the VINT register as indicated in the following table:

Table D-6 VMEbus Interrupt Level Encoding

VMEbus Interrupt Level	IL2-IL0 Setting
7	111
6	110
5	101
4	100
3	011
2	010
1	001
0	(no interrupt)

The corresponding interrupt vector supplied upon IACK is programmed into the lower 8-bits of the IVECT register.

IVECT Register (0x7d00.0024)



Any reset will return the SCV64 registers to their default values. Thus all VMEbus interrupts will be deasserted.

D.7 VME Interrupt Handling

Caution – The 5/64 implements custom interrupt handling and acknowledgment. Please refer to *VMEbus/SCV64 Interrupts on page 6-69* for details on the implementation and programming of the interrupt handler. The following SCV64 material is presented for reference only

The SCV64 interrupt handler module accepts interrupt requests from three groups of sources listed according to their priority below:

1. Level 7 interrupts
2. General local interrupts
3. VMEbus interrupts

All interrupts are channeled to a common interrupt handler and all interrupt acknowledge cycles are processed by a single acknowldgement module.

Each interrupt group has its own control register for enabling/disabling. The 7IE register controls level 7 interrupts, the LIE register controls generla local interrupts and the VIE register enables VMEbus interrupts. An interrupt source is enabled by setting its corresponding control bit in its control register. All interrupts (except L7INMI - which is always enabled) are reset by disabling and re-enabling them through their respective control bits. L7INMI is reset by clearing the NMICLR bit in the 7IE register

Table D-7 Control Bits for Interrupt Sources

Level 7 Interrupts	Control Bits in 7IE Register 0x7d00.00A8	General Local Interrupts	Control Bits in LIE Register 0x7d00.00AC	VMEbus Interrupts	Control Bits in VIE Register 0x7D00.00BO
*L7NMI	NMICLR	LIRQ5	L5E	IRQ7	V7E
*L7IMEM	MEMIE	LIRQ4	L4E	IRQ6	V6E
*L7IACF	ACFIE	LIRQ3	L3E	IRQ5	V5E
BIMODE	BIE	LIRQ2	L2E	IRQ4	V4E
*SYSFAIL	SYSFIE	LIRQ1	L1E	IRQ3	V3E
		LIRQ0	L0E	IRQ2	V2E
				IRQ1	V1E

D.8 Location Monitor

The Location Monitor resides at the top longword (32 bits) of each A32 and A24 slave image and is accessible from both the local and VMEbus. The bottom (even) word of the Location Monitor is used for 16-bit messages. Since the Location Monitor replaces the long word of memory which otherwise exists at that address, writes to the Location Monitor are not turned into local cycles or entered into the RXFIFO. In addition, read accesses to the Location Monitor result in a bus error.

Writes to the the Location Monitor are queued in the LMFIFO (32 bits wide and 31 stages deep) and read from the LMFIFO register. While there are entries in the LMFIFO, the SCV64 asserts LMINT and sets the LMHD bit in the DCSR register. Since LMINT is only asserted while there are entries in the LMFIFO, an interrupt service routine or polling of the LMINT signal can be used to test for entries in the LMFIFO and support various Location Monitor functions.

The CPU may write to its own Location Monitor through its slave image as it would any address on the VMEbus. The SCV64 address decoder responds by redirecting the message to the LMFIFO. From there the message is handled in the same manner as a write from the VMEbus side. If the condition occurs in

which the local CPU and VMEbus write to the Location Monitor at the same time, the race condition is resolved and the writes queued without loss of either of the message providing space is available in the LMFIFO).

One function of the Location Monitor is as a command to exit BI-mode. When there is a write to the Location Monitor, the SCV64 is removed from BI-mode and the BIMODE signal is cleared.

LMFIFO Location Monitor Read Port (0x7D00.0038)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Location Monitor FIFO Output Stage - Byte 3								Location Monitor FIFO Output Stage - Byte 3							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Location Monitor FIFO Output Stage - Byte 3								Location Monitor FIFO Output Stage - Byte 3							

D.9 DMA Transfers

The SCV64 performs all BLT and MBLT cycles through the DMA controller. During DMA transfers, the SCV64 is both the VME and local master. In case of a DMA write operation, the source address is contained in the DMA Local Address Register (DMALAR).

DMALAR DMA Local Address Register (0x7d00.0000)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Not Used					DLA (DMA Local Addr)			DLA (DMA Local Address) - Byte 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DLA (DMA Local Address) - Byte 1								DLA (DMA Local Address) - Byte 0							0

DMA reads use the DMA VME Address Register (DMAVAR) for the source address.

DMAVAR DMA VMEbus Address Register (0x7D00.0004)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DVA (DMA VME Address) - Byte 3								DVA (DMA VME Address) - Byte 2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVA (DMA VME Address) - Byte 1								DVA (DMA VME Address) - Byte 0							

Data from the source address is transferred to the RXFIFO or TXFIFO for DMA read and write operations respectively.

The various addressing and data transfer modes applicable to the DMA operation are selected using the MODE register. The required bit settings for the supported addressing (A64, A32, A24) and data modes (D32, D16) are indicated below.

For A64 transfers, the MA64BAR register is used for the upper 32-bits of the source or destination address.

The size of the DMA transfer is controlled by both the DMATC register and the DTCSIZ bit in the MODE register. With the DTCSIZ bit cleared (default), the DMATC register will count up to 4096 32-bit transfers (16KB). If the DTCSIZ bit is set, the DMATC register will count up to 1,048,576 transfer (4MB).

DMATC DMA Transfer Count Register (0x7D00.0008)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Not Used								Not Used				DTC (DMA Transfer Count)			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTC (DMA Transfer Count								DTC (DMA Transfer Count							

The transfer count for VMEbus cycles is recorded in the DMAVTC register. The difference between the value in the DMATC register and the DMAVTC is equal to the number of entries in the RXFIFO or TXFIFO.

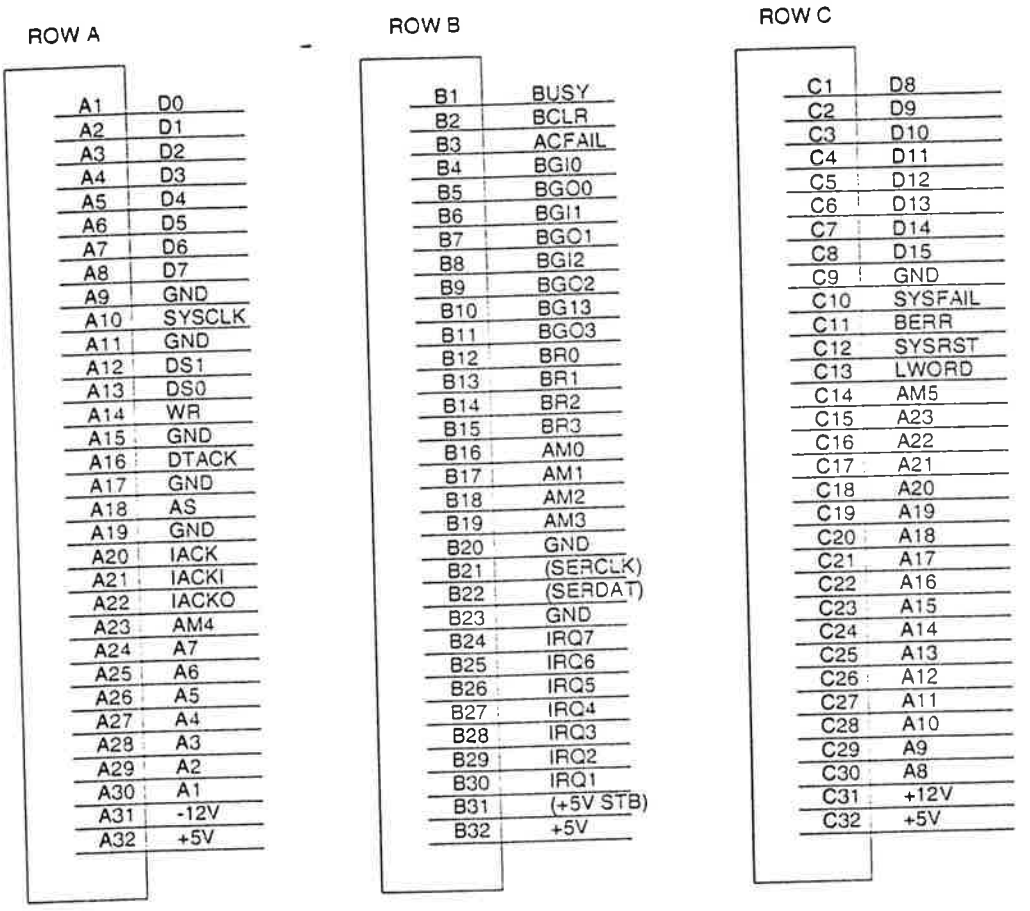
DMAVTC DMA VMEbus Transfer Count Register(0x7D00.004C)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Not Used								Not Used				DMA VME Transfer Count			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA VME Transfer Count								DMA VME Transfer Count							

5/64 Connector Specifications

In this appendix is a complete listing of all connector types specifically for the Themis SPARC 5/64 board.

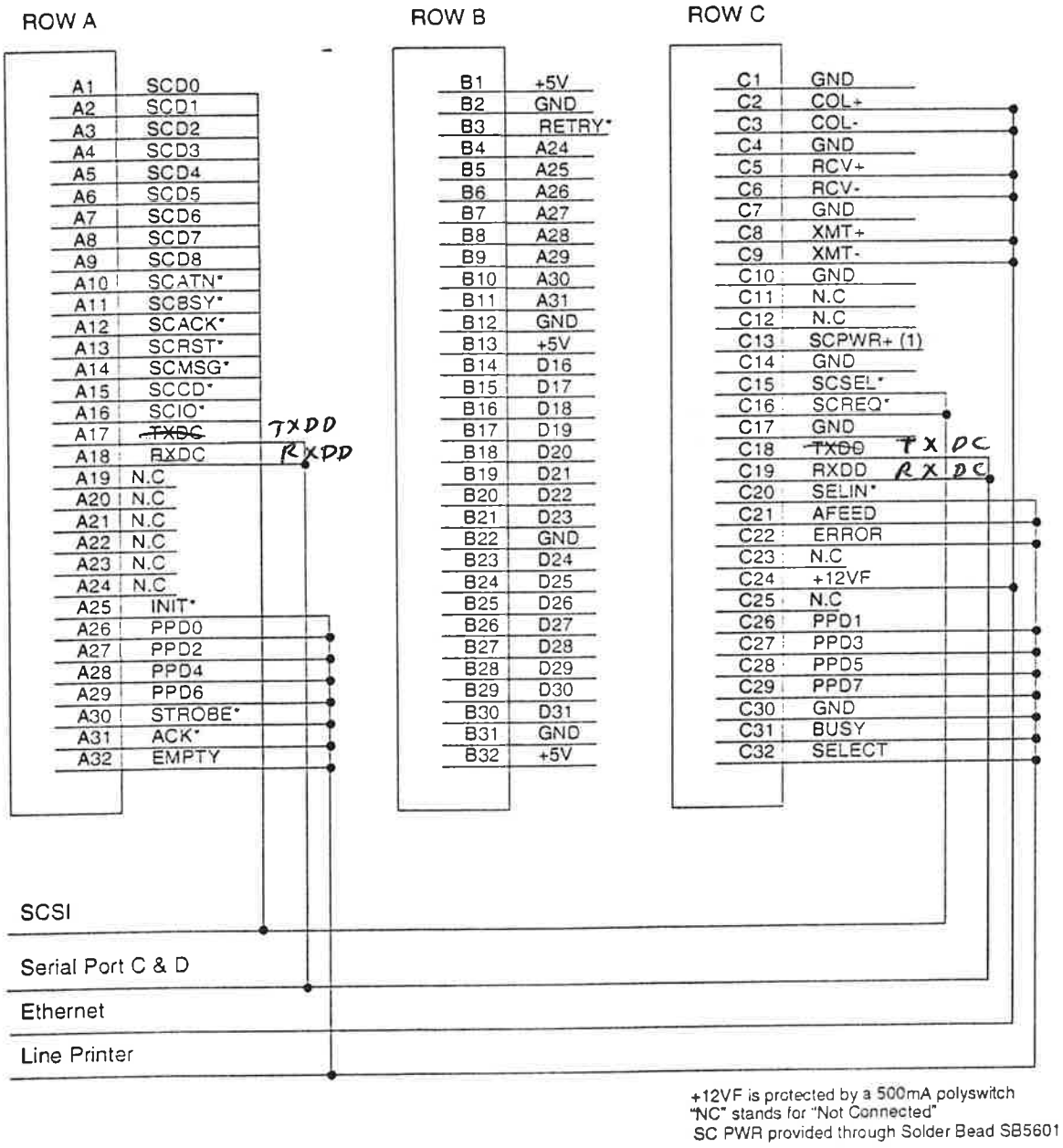
E.1 VME Bus Connectors (P1)



ROW A		ROW B		ROW C	
A1	D0	B1	BUSY	C1	D8
A2	D1	B2	BCLR	C2	D9
A3	D2	B3	ACFAIL	C3	D10
A4	D3	B4	BGI0	C4	D11
A5	D4	B5	BGO0	C5	D12
A6	D5	B6	BGI1	C6	D13
A7	D6	B7	BGO1	C7	D14
A8	D7	B8	BGI2	C8	D15
A9	GND	B9	BGO2	C9	GND
A10	SYSCLK	B10	BG13	C10	SYSFAIL
A11	GND	B11	BGO3	C11	BERR
A12	DS1	B12	BR0	C12	SYSRST
A13	DS0	B13	BR1	C13	LWORD
A14	WR	B14	BR2	C14	AM5
A15	GND	B15	BR3	C15	A23
A16	DTACK	B16	AM0	C16	A22
A17	GND	B17	AM1	C17	A21
A18	AS	B18	AM2	C18	A20
A19	GND	B19	AM3	C19	A19
A20	IACK	B20	GND	C20	A18
A21	IACKI	B21	(SERCLK)	C21	A17
A22	IACKO	B22	(SERDAT)	C22	A16
A23	AM4	B23	GND	C23	A15
A24	A7	B24	IRQ7	C24	A14
A25	A6	B25	IRQ6	C25	A13
A26	A5	B26	IRQ5	C26	A12
A27	A4	B27	IRQ4	C27	A11
A28	A3	B28	IRQ3	C28	A10
A29	A2	B29	IRQ2	C29	A9
A30	A1	B30	IRQ1	C30	A8
A31	-12V	B31	(+5V STB)	C31	+12V
A32	+5V	B32	+5V	C32	+5V

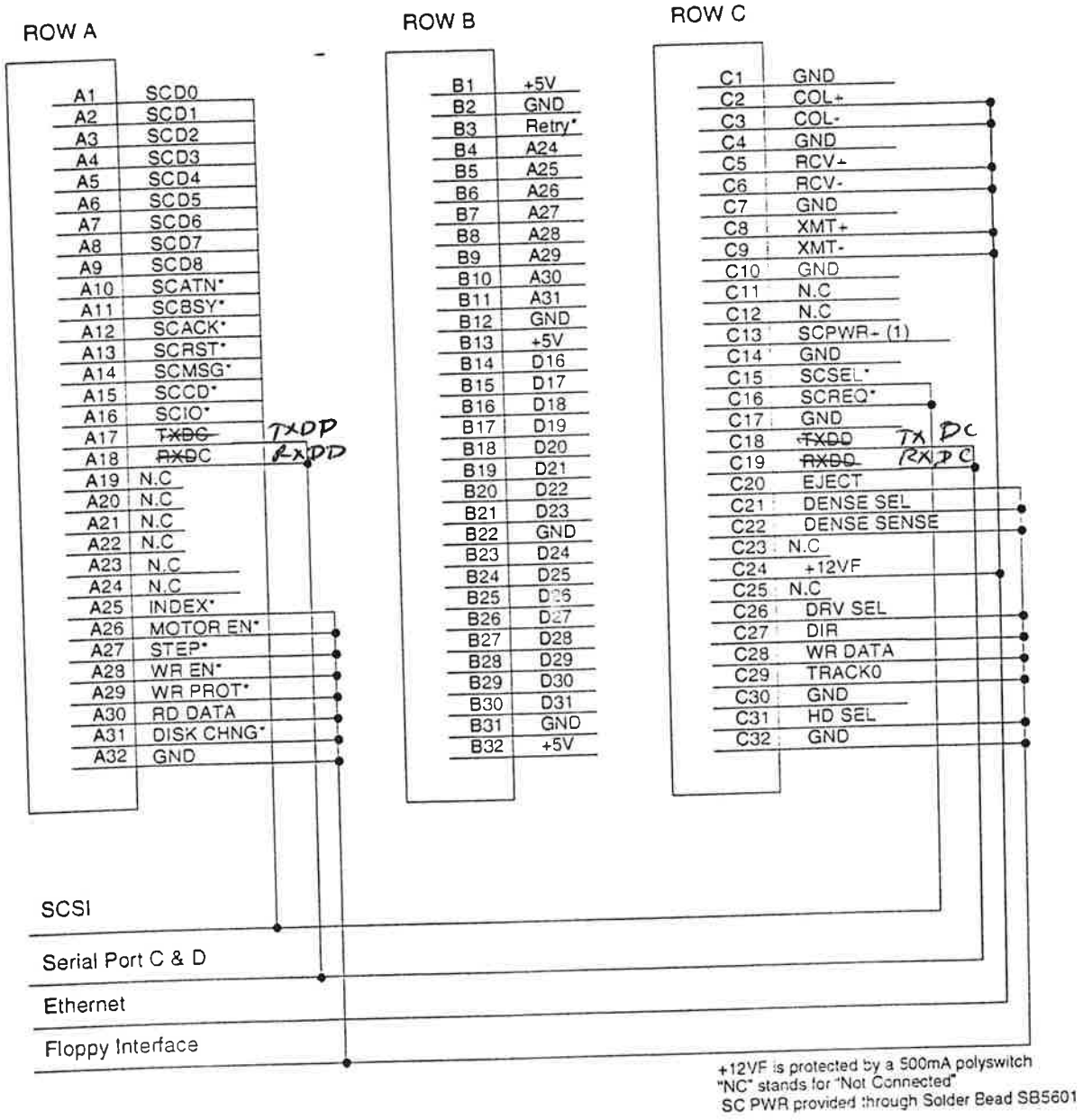
"NC" stands for "Not Connected"

E.2 *VME Bus Printer Interface Connectors(P2) Factory Standard*



+12VF is protected by a 500mA polyswitch
"NC" stands for "Not Connected"
SC PWR provided through Solder Bead SB5601

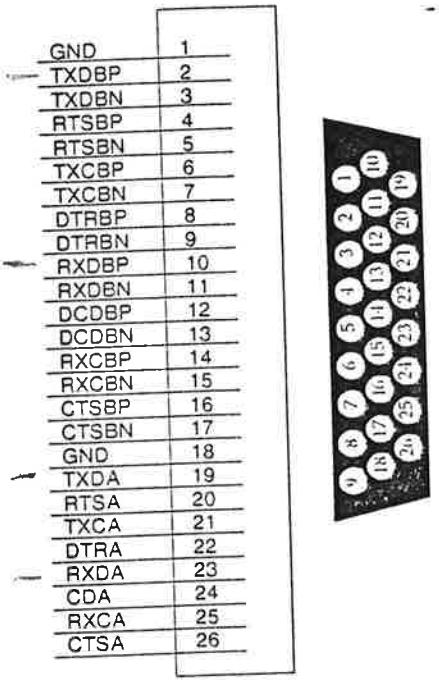
E.3 VME Bus Floppy Interface Connectors Continued (P2) Floppy Port Enabled



E.4 SBus Connectors Pinout:J602 (Slot 1) and J601(Slot 0)

GND	1	49	SCLK
BR0/1	2	50	BG0/1
SEL0/1	3	51	SBAS0/1
SINT1	4	52	GND
SD0	5	53	SD1
SD2	6	54	SD3
SD4	7	55	SD5
SINT2	8	56	+5V
SD6	9	57	SD7
SD8	10	58	SD9
SD10	11	59	SD11
SINT3	12	60	GND
SD12	13	61	SD13
SD14	14	62	SD15
SD16	15	63	SD17
SINT4	16	64	+5V
SD19	17	65	SD18
SD21	18	66	SD20
SD23	19	67	SD22
SINT5	20	68	GND
SD25	21	69	SD24
SD27	22	70	SD26
SD29	23	71	SD28
SINT6	24	72	+5V
SD31	25	73	SD30
SZ0	26	74	SZ1
SZ2	27	75	SBRO
SINT7	28	76	GND
SA0	29	77	SA1
SA2	30	78	SA3
SA4	31	79	SA5
LERR	32	80	+5V
SA6	33	81	SA7
SA8	34	82	SA9
SA10	35	83	SA11
ACK0	36	84	GND
SA12	37	85	SA13
SA14	38	86	SA15
SA16	39	87	SA17
ACK1	40	88	+5V
SA18	41	89	SA19
SA20	42	90	SA21
SA22	43	91	SA23
ACK2	44	92	GND
SA24	45	93	SA25
SA26	46	94	SA27
NC	47	95	SBRST*
-12V	48	96	+12V

E.5 Serial Ports A and B: 26-pin Connector (P3001)

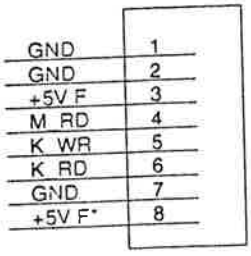


Name	Dir	Description
TXD	Out	Transmitted Data from 5/64
RXD	In	Received Data to by Terminal
CTS	In	Clear To Send
RTS	Out	Request To Send
GND	-	Signal Ground
DTR	Out	Data Terminal Ready
DCD	In	Data Carrier Detect
Transmit Clk	Out	Transmit Clock (synchronous mode)
Receive Clk	In	Receive Clock (synchronous mode)

Terminology: Out means "from 5/64 to Terminal"
In means "from Terminal to 5/64"

Mating Connector Part Number: #748365-1
22-28 gauge wire, contact #748333-4

E.6 Keyboard/Mouse: 8-Pin DIN Connector (P3002)



*+5VF: protected by 500mA Polyswitch

Mating Connector Part Number: Shell #84165-2, Housing #84168-1, & Contacts #787257-1

E.7 SCSI Connector: High-Density 50-pin (P2401)

SCD0	26	1	GND
SCD1	27	2	GND
SCD2	28	3	GND
SCD3	29	4	GND
SCD4	30	5	GND
SCD5	31	6	GND
SCD6	32	7	GND
SCD7	33	8	GND
SCDP	34	9	GND
NC	35	10	GND
NC	36	11	GND
NC	37	12	GND
TRMPWR	38	13	GND
NC	39	14	GND
NC	40	15	GND
SCATN	41	16	GND
NC	42	17	GND
SCBSY	43	18	GND
SCACK	44	19	GND
SCRST	45	20	GND
SCMSG	46	21	GND
SCSEL	47	22	GND
SCCD	48	23	GND
SCREQ	49	24	GND
SCIO	50	25	GND

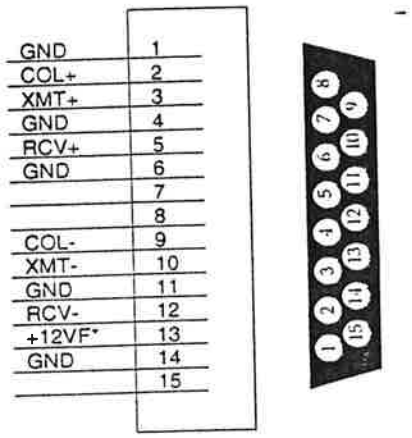
Mating Connector Part Number: #1-747285-0

E.8 10 Base T: RJ45 Twisted-Pair Connector (P2301)

TP OUT+	1
TP OUT-	2
TP IN+	3
TP IN-	6
NC	4
NC	5
NC	7
NC	8

Mating Connector Part Number: #5-554739-2

E.9 AUI Ethernet Connector (P2302)



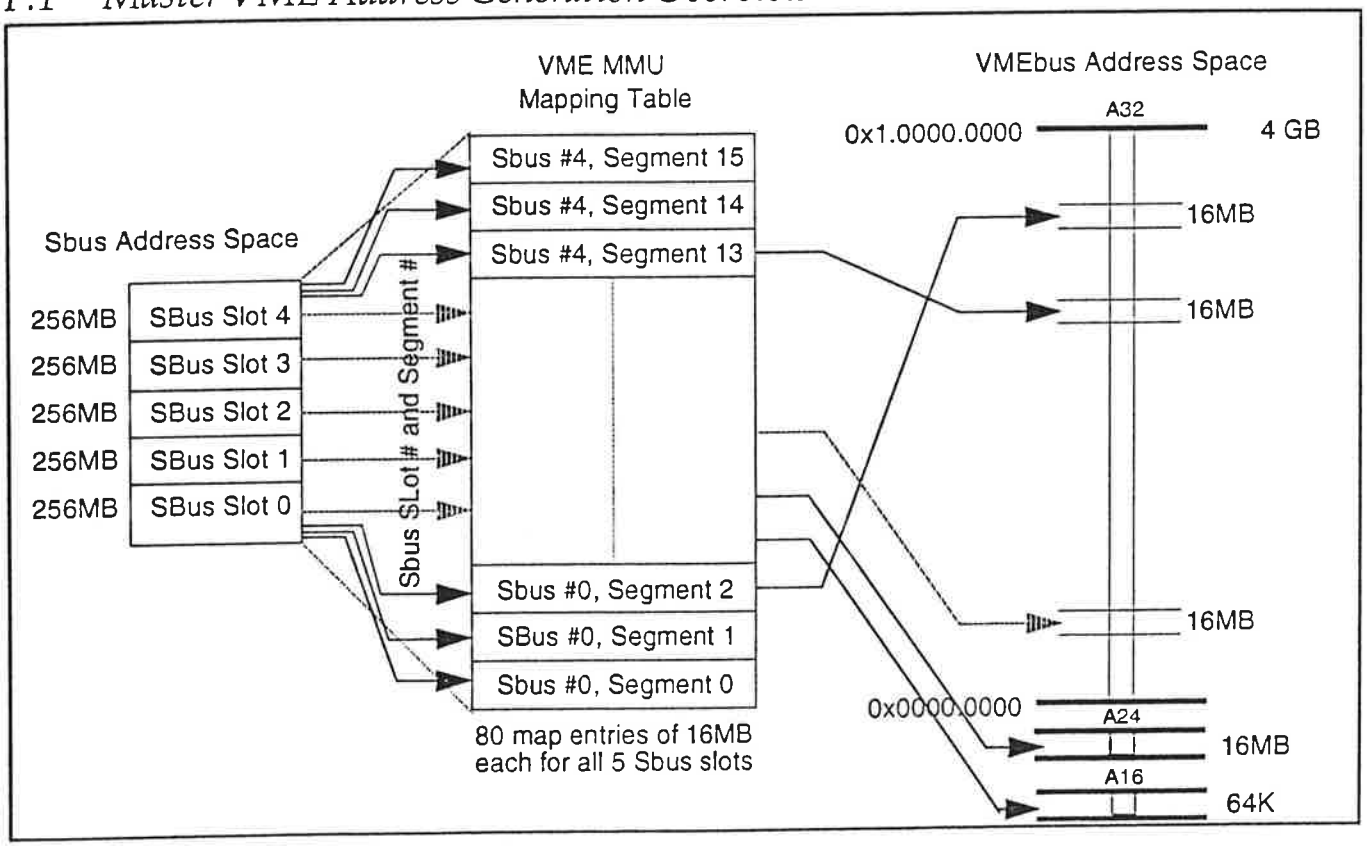
*+12VF protected by 500mA Polyswitch

Mating Connector Part Number: #205203-3
24/20 gauage wire, contacts #66504-9

Pin	Name	Dir	Description
1	GND		
2	COL+	In	Ethernet Collision Line (+)
3	XMT+	Out	Ethernet Transmit Data (+)
4	GND		
5	RCV+	In	Ethernet Received Data (+)
6	GND		
7			
8			
9	COL-	In	Ethernet Collision Line (-)
10	XMT-	Out	
11	GND		
12	RCV-	In	Ethernet Received Data (-)
13	+12VF	PWR	Fused +12V
14	GND		
15			

Address Mapping Specifications

F.1 Master VME Address Generation Overview

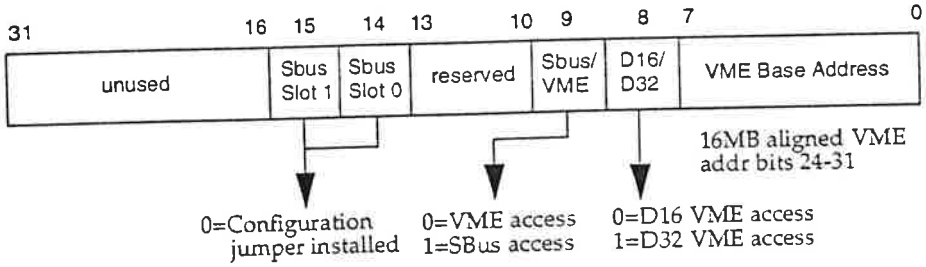


F.2 *Master VME MMU and SBus Slot Address Mapping*

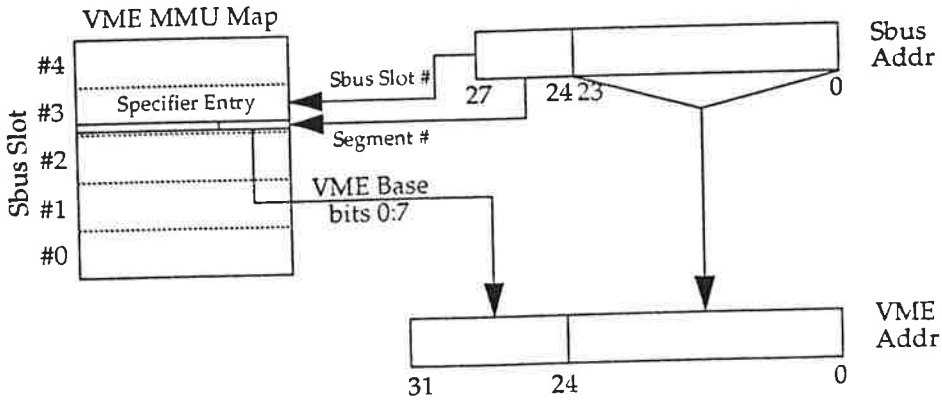
The microSPARC II SPARC processor implementation is an SBus based design. To address resources (aka memory locations) on the VMEbus, SBus addresses must be mapped to the VMEbus.

The 5/64 implements flexible mapping of SBus resources to local or VMEbus resources with a VME MMU map translation table.

For the purpose of the VME MMU table, each 256MB SBus slot address space of SBus slots #0 through #4 is divided into 16 identical segments of 16MB. This design yields a VME MMU map of 80 entries of 32-bits, each defined as:



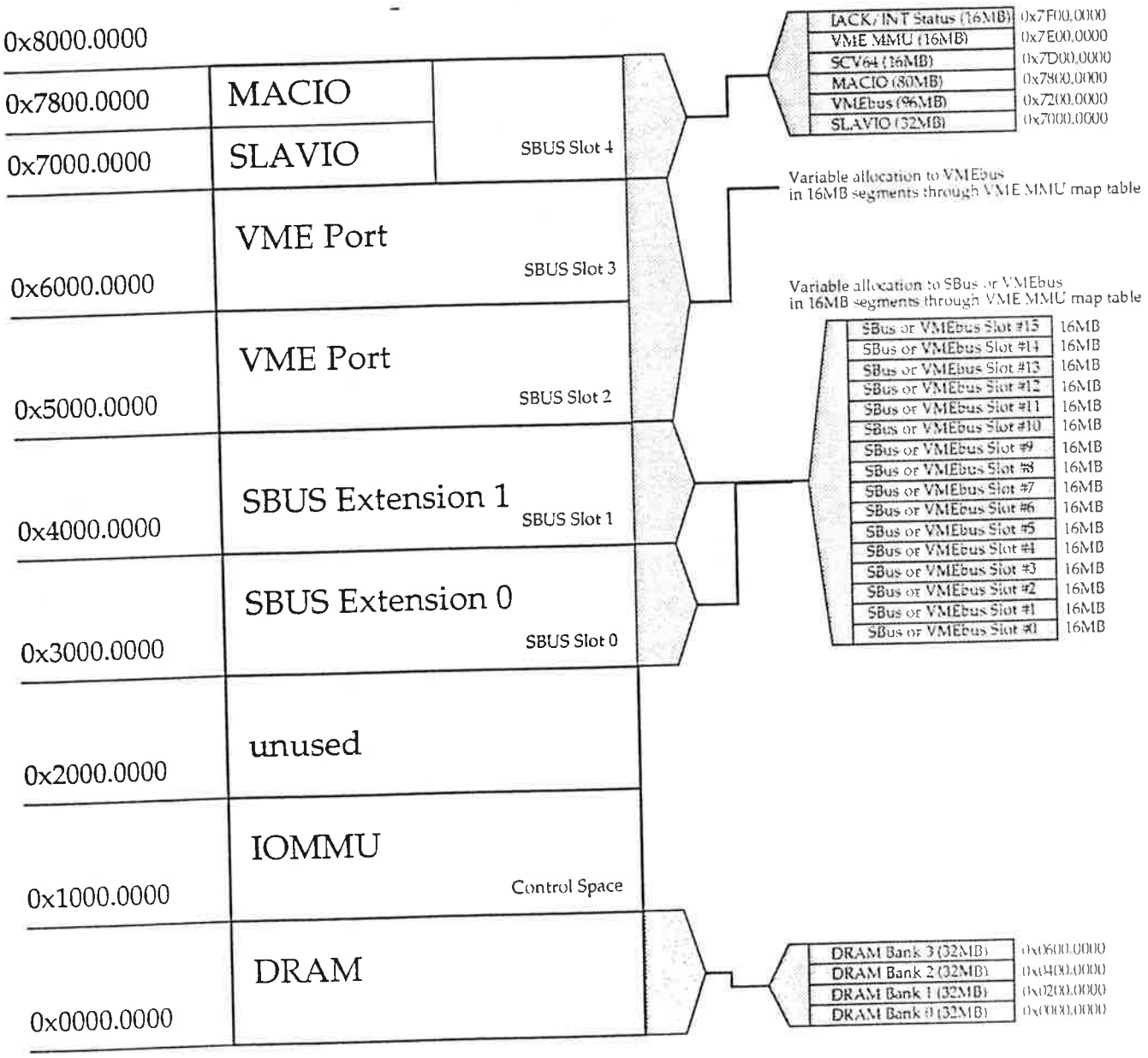
Any SBus access is translated by the VME MMU. Based on bit 9, the access is either an actual SBus access (local resource) or VME resources. In that later case, it is converted into a corresponding VMEbus access.



Certain SBus slot number and segment offset combinations are hardcoded to specific, pre-defined mapping relations and cannot be altered. Mappings on SBus slots 0 and 1 also depend on the configuration of solder beads F3501 and F3502. The following table gives an overview of the default mapping, including the fixed translation entries (hardcoded entries are shaded for emphasize):

Segment	SBus #0	SBus #1	SBus #2	SBus #3	SBus #4
0	A32 or SBUS	A32 or SBUS	A32	A32	SLAVIO
1	A32 or SBUS	A32 or SBUS	A32	A32	SLAVIO
2	A32 or SBUS	A32 or SBUS	A32	A32	A32
3	A32 or SBUS	A32 or SBUS	A32	A32	A32
4	A32 or SBUS	A32 or SBUS	A32	A32	A32
5	A32 or SBUS	A32 or SBUS	A32	A32	A24:D32
6	A32 or SBUS	A32 or SBUS	A32	A32	A24:D16
7	A32 or SBUS	A32 or SBUS	A32	A32	A16:D16
8	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
9	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
10	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
11	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
12	A32 or SBUS	A32 or SBUS	A32	A32	MACIO
13	A32 or SBUS	A32 or SBUS	A32	A32	SCV64
14	A32 or SBUS	A32 or SBUS	A32	A32	VMEMMU
15	A32 or SBUS	A32 or SBUS	A32	A32	IACK

F.3 *microSPARC II Mapping*



F.4 *DRAM Addressing Window*

The DRAM address space of the microSPARC II processor is divided into 8 DRAM-bank (bank0 through bank7) of 32MB each.

DRAM bank0 is on-board memory of either 8MB or 32MB size. DRAM banks 1 and 3 can be mounted on the two #1 SIMM sockets (IC301 and IC302) and bank #2 can be mounted on the two #2 SIMM sockets (IC303 and IC304).

The SIMM sockets #1 support 4MB, 16MB and 32MB of Sun compatible memory modules. SIMM sockets #2 only support 4MB or 16MB Sun compatible memory modules.

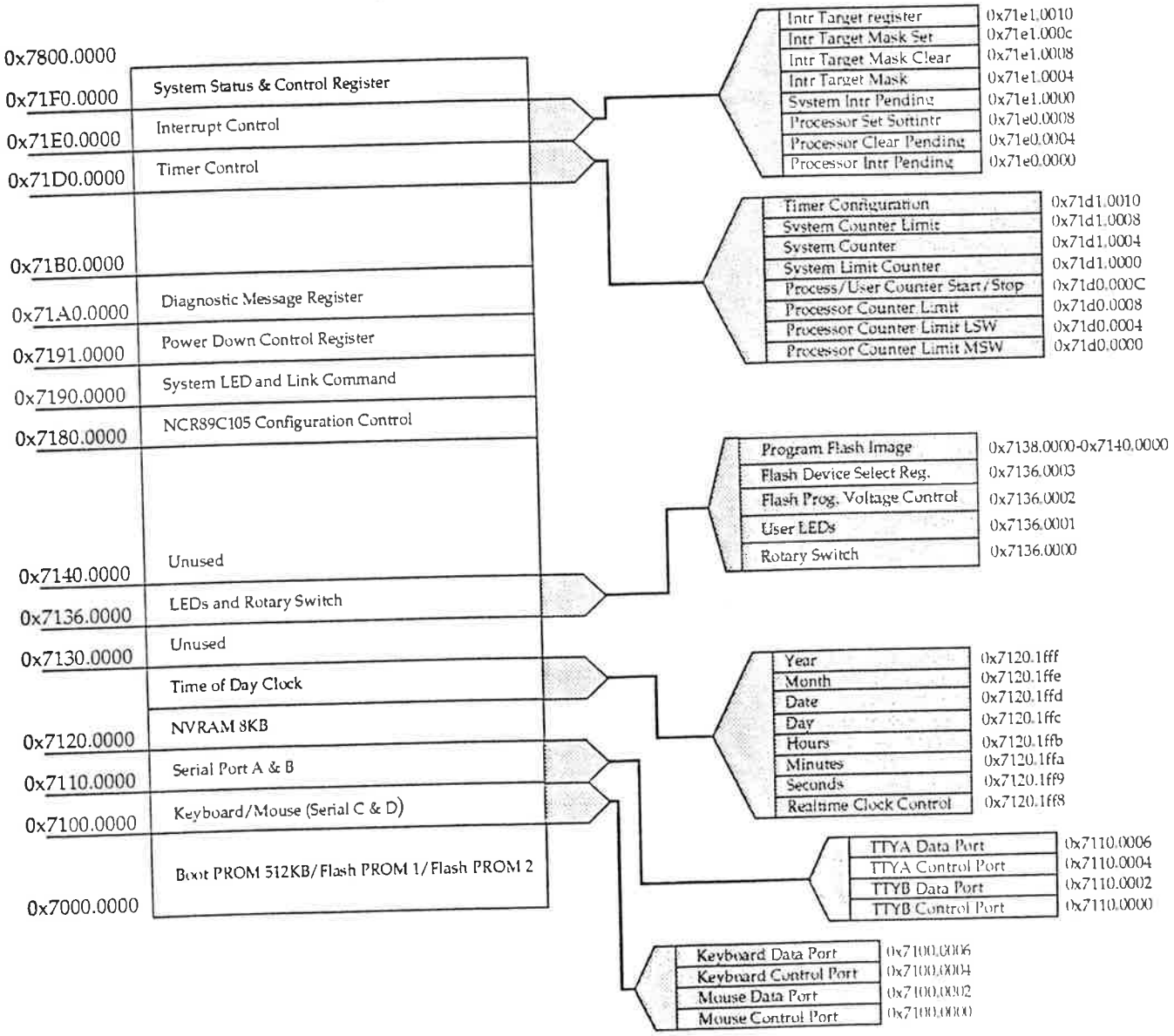
Caution – Due to physical design limitations, installation of SIMM modules in bank #2 and/or #3 prevent use of an SBus card in SBus slot 0.

All possible 5/64 memory configuration options are listed below:

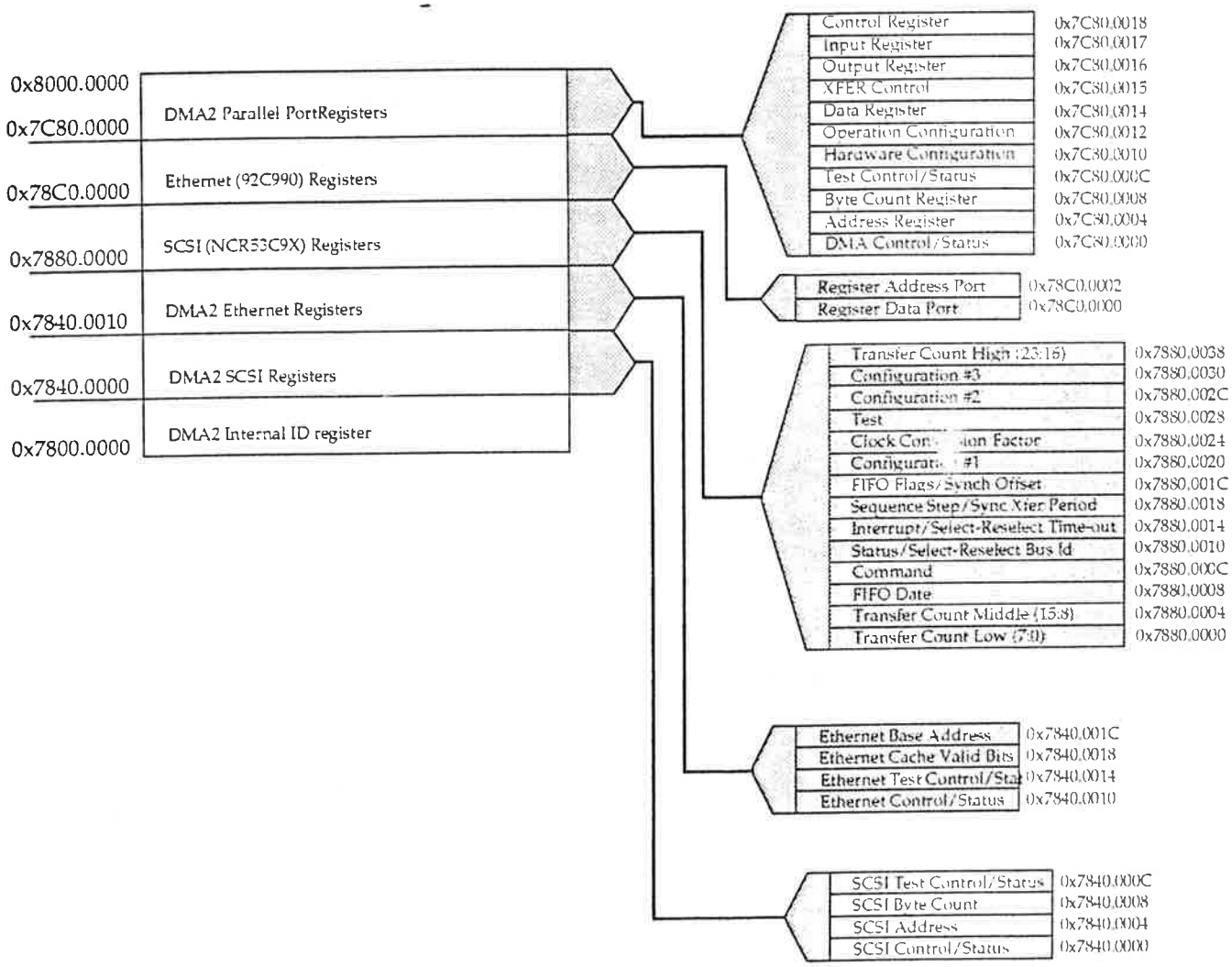
Capacity	Onboard	SIMM #1	SIMM #2	SLOT #0	SLOT #1
8MB	8MB	0	0	Yes	Yes
16MB	8MB	8MB	0	Yes	Yes
40MB	8MB	32MB	0	Yes	Yes
72MB	8MB	64MB	0	No	Yes
16MB	8MB	0	8MB	No	Yes
24MB	8MB	8MB	8MB	No	Yes
48MB	8MB	32MB	8MB	No	Yes
80MB	8MB	64MB	8MB	No	Yes
40MB	8MB	0	32MB	No	Yes
48MB	8MB	8MB	32MB	No	Yes
72MB	8MB	32MB	32MB	No	Yes
104MB	8MB	64MB	32MB	No	Yes
32MB	32MB	0	0	Yes	Yes
40MB	32MB	8MB	0	Yes	Yes
64MB	32MB	32MB	0	Yes	Yes
96MB	32MB	64MB	0	No	Yes
40MB	32MB	0	8MB	No	Yes
48MB	32MB	8MB	8MB	No	Yes
72MB	32MB	32MB	8MB	No	Yes
104MB	32MB	64MB	8MB	No	Yes
64MB	32MB	0	32MB	No	Yes
72MB	32MB	8MB	32MB	No	Yes
96MB	32MB	32MB	32MB	No	Yes
128MB	32MB	64MB	32MB	No	Yes

F.5 SLAVIO/EBus Mapping

The SLAVIO-controlled EBus connects all non-DMA 8-bit peripheral devices



F.6 MACIO Mapping



F.7 Rotary Switch Register

Access Mode: BYTE Read Only

Base Address: 0x7136.0000

Bit #	Description
0-3	Rotary Switch Value
4-7	Undefined

This register returns the current two's complement value of the front panel rotary switch.

F.8 User LED Register

Access Mode: BYTE Write Only

Base Address: 0x7136.0001

This register controls the on/off condition of the 4 user LEDs on the front panel.

F.9 FLASH Programming Voltage Control Register

Access Mode: BYTE Read/Write

Base Address: 0x7136.0002

Bit #	Description
0	FLASH Programming Voltage: 0 = Remove programming voltage 1 = Apply programming voltage to FLASH device selected
1	Boot PROM Device Selection: 0 = Boot PROM device is UVPROM 1 = Boot PROM device is FLASH PROM

Bit #	Description
2_	FLASH Programming Enable/Disable: 0 = FLASH programming disabled (jumper J4700 installed) 1 = FLASH programming enabled (jumper J4700 removed)
3-7	No Effect

This register control enabling and disabling the programming voltage for the FLASH EPROM devices. A delay of 10ms is necessary after enabling the programming voltage and commencing actual FLASH programming

Note: The programming is dependent on the flash type (Manufacturer code and device code).

F.10 FLASH Select Register

Access Mode: Byte Read/Write

Base Address: 0x7136.0003

Bit #	Description
0	FLASH Device Selection For Programming: 0 = select FLASH EPROM #1 1 = select FLASH EPROM #2
1-7	No Effect

This register selects one of the two FLASH EPROM registers. Only the FLASH device selected can be programmed. Bit 0 is cleared during RESET.

F.11 KSIC/SCV64 Interrupt Status Register

Access Mode: 8-bit or 32-bit Word Read Only

Base Address: 0x7F0F.0400 (32-bit) or 0x7F0F.0403 (byte)

Bit #	Description
0	Undefined
1	0 = SCV64 IRQ1 inactive 1 = SCV64 IRQ1 pending
2	0 = SCV64 IRQ2 inactive 1 = SCV64 IRQ2 pending
3	0 = SCV64 IRQ3 inactive 1 = SCV64 IRQ3 pending
4	0 = SCV64 IRQ4 inactive 1 = SCV64 IRQ4 pending
5	0 = SCV64 IRQ5 inactive 1 = SCV64 IRQ5 pending
6	0 = SCV64 IRQ6 inactive 1 = SCV64 IRQ6 pending
7	0 = SCV64 IRQ7 inactive 1 = SCV64 IRQ7 pending
8-31	No Effect

F.12 VME Interrupt Acknowledge Registers

Access Mode: 8-bit or 32-bit Word Read Only

Base Address: 0x7F0F.0000 (32-bit) 0x7F0F.0003 (byte)

0x7F0F.001C	Interrupt Acknowledge Register VMEIRQ7
0x7F0F.0018	Interrupt Acknowledge Register VMEIRQ6
0x7F0F.0014	Interrupt Acknowledge Register VMEIRQ5
0x7F0F.0010	Interrupt Acknowledge Register VMEIRQ4
0x7F0F.000C	Interrupt Acknowledge Register VMEIRQ3
0x7F0F.0008	Interrupt Acknowledge Register VMEIRQ2
0x7F0F.0004	Interrupt Acknowledge Register VMEIRQ1

To acknowledge VMEbus interrupts received by the SCV64 controller and pass them to the SBus, a set of 7 interrupt acknowledge registers are implemented.

The base address of these registers is 0x7F0F.0000 and VMEbus interrupt level n generated by the SCV64 is acknowledge by the CPU reading a 32-bit value from the location computed by:

```
register address = 0x7F0F.0000 + 4 * n (32-bit access)
                 = 0x7F0F.0003 + 4 * n (8-bit access)
```

where n is a value between 1 and 7.

Caution – If an attempt is made to acknowledge an interrupt that is not pending, a bus error is returned.

F.13 *Local SCV64 Interrupt Acknowledge Registers*

Access Mode: 8-bit or 32-bit Word Read Only

Base Address: 0x7F0F.0400 (32-bit) 0x7F0F.0403 (byte)

0x7F0F.041C	Interrupt Acknowledge Register LIRQ7
0x7F0F.0418	Interrupt Acknowledge Register LIRQ6
0x7F0F.0414	Interrupt Acknowledge Register LIRQ5
0x7F0F.0410	Interrupt Acknowledge Register LIRQ4
0x7F0F.040C	Interrupt Acknowledge Register LIRQ3
0x7F0F.0408	Interrupt Acknowledge Register LIRQ2
0x7F0F.0404	Interrupt Acknowledge Register LIRQ1

To acknowledge local interrupts received by the SCV64 controller and pass them to the SBus, a set of 7 interrupt acknowledge registers is implemented.

The base address of these registers is 0x7F0F.0400 and interrupt level n generated by the SCV64 is acknowledge by the CPU reading a 32-bit value from the location computed by:

```
register address = 0x7F0F.0400 + 4 * n (32-bit access)
                 = 0x7F0F.0403 + 4 * n (8-bit access)
```

where n is a value between 1 and 7.

Caution – If an attempt is made to acknowledge an interrupt that is not pending, a bus error is returned.

F.14 VMMU Enable Register

Access Mode: 32-bit Word Read/Write

Base Address: 0x7F0F.0000

Bit #	Description
0	VMMU Enable/Disable Bit: Read access : 0 = VMMU translations disabled 1 = VMMU translations enabled Write access: Toggle
1-7	Undefined

Upon each write access, this register toggles the state of bit #0, alternately enabling or disabling the VMMU translation mechanism. A read access returns the current status of bit #0 without altering it.

5/64 Jumper Settings

G.1 Factory Default Settings

Id	Type	Default	Description
SB701	Bead	Installed	Enable front panel RESET switch
IC3008	Bead	On IC3008B: 1-10,3-8,5-6 On IC3008A: 2-9,4-7	Serial channel B configured as RS-232
J3001	Jumper	2-3	Data from front panel keyboard connector
J3002	Jumper	2-3	Data from front panel mouse connector
F4500	Bead	Varies	Select +5V/+12V or +5V only
F5500	Bead	Installed	VMEBus BERR connected SCV64
J2401	Jumper	1-2	SCSI active termination enabled
F3501	Bead	Installed (KSIC)	SBUS slot 0 allocated entirely SBUS
F3502	Bead	Installed (KSIC)	SBUS slot 1 allocated entirely SBUS
J4700	Jumper	Installed (KSIC)	FLASH programming disabled

Id	Type	Default	Description
F6500 F6501	Beads	1-2	Memory bank #0 mapping set to 0x0000.0000
F2501 F2502	Beads	Removed (KSIC)	Slave window size 8MB
J2001	Jumper	1-2	VME system controller enabled
J2101	Jumper	Removed	VMEBus enabled
SB6500 SB6501	Bead	Varies (@85MHz off) (@85MHz on)	Memory Speed Select
SB6502 SB6503		(@85MHz off) (@85MHz off)	
F6600	Bead	Varies (@85MHz on)	CPU Clock Select
SB5601	Bead	Removed	SCSI termination power disconnected from P2-C13
SB6501	Bead	1-2	Printer interface on P2 (17 beads)

G.2 5/64 Jumper Locations and Configuration

The 5/64 includes 8 user configurable jumpers that control various aspects of the SBCs behavior as well as a number of solder beads for less frequently modified parameters.

The SCV64 VMEbus interface chip can be configured to provide VME system controller functions. When enabled, it will activate the arbiter and other VMEbus monitoring functions.

Serial ports C & D can be routed to the front panel (functioning as keyboard/mouse ports) or the paddle board via the P2 connector.

Serial port B can be selectively configured as RS232 or RS422 interface.

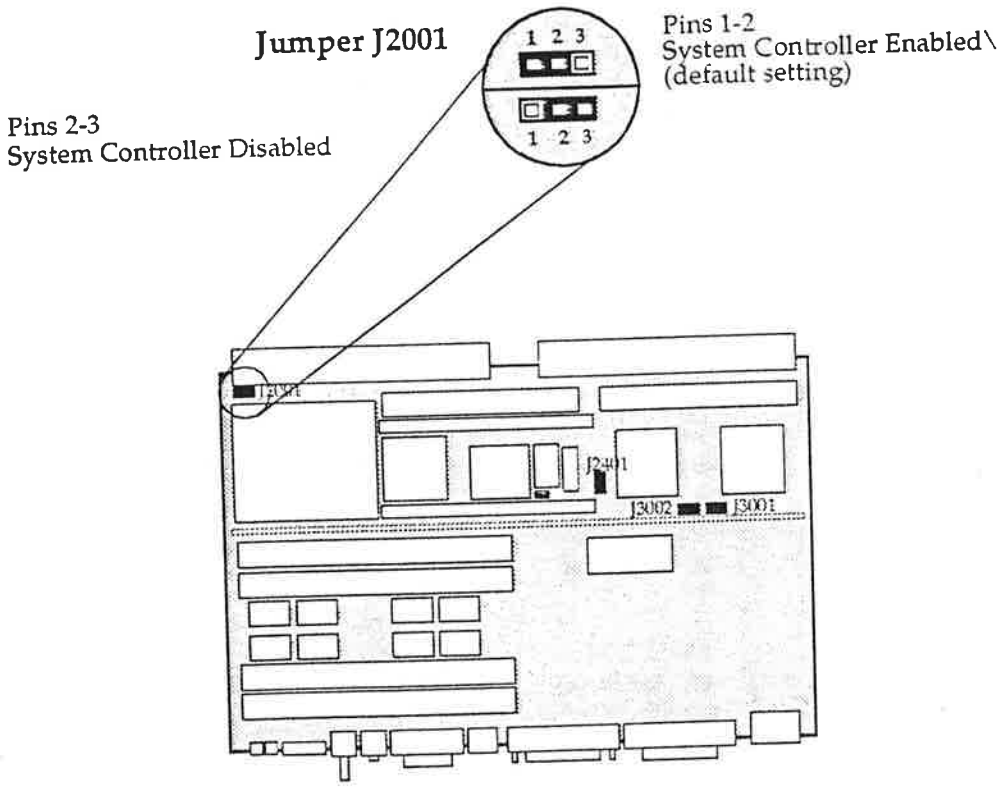
With SCSI-2 interfaces provided at both the front panel and P2, the 5/64 may be located within a SCSI chain. A jumper is provided to disable SCSI termination if the 5/64 is not a terminal node on the SCSI bus.

FLASH EPROM programming voltages of +5V/+12V or +5V only can be selected via solder beads, depending on the FLASH EPROM type installed. The default factory configuration for this bead is FLASH dependant and should not be modified unless the installed FLASH EPROM is replaced with a different type. Additionally, FLASH EPROM prpgramming can be enabled or disabled via a jumper. (J4700 on KSIC).

A solder bead allows users to physically disconnect the VME BERR signal from the SCV64 controller. Another jumpers can be used to completely disconnect the SBC from the VMEbus.

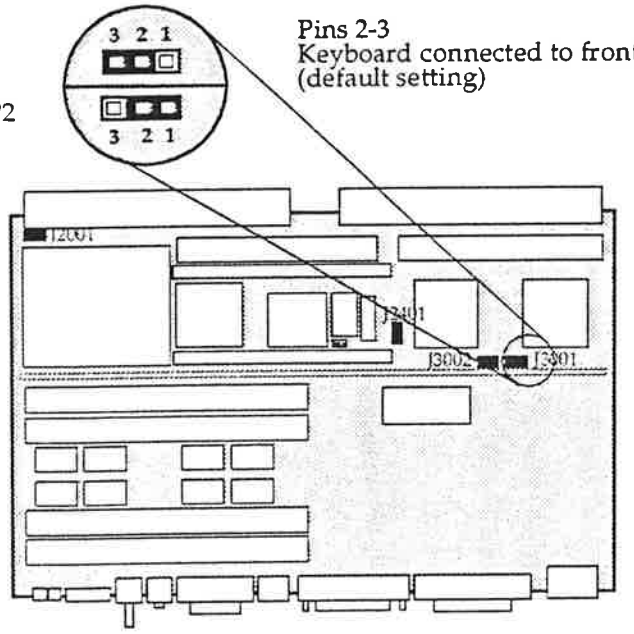
Two solder beads allow configuration of SBUS slots 0 and 1 as SBUS only address slots or VMEbus/SBus address slots controlled by the VMMU.

Due to signal limitations on the P2 connector, solder beads are used to select availability of either the floppy disk or parallel printer interface on the VME P2 connector.


Jumper J2001
1 2 3
1 2 3
Pins 1-2
System Controller Enabled\
(default setting)
Pins 2-3
System Controller Disabled
J2401
J3002
J3001

Jumper J3001

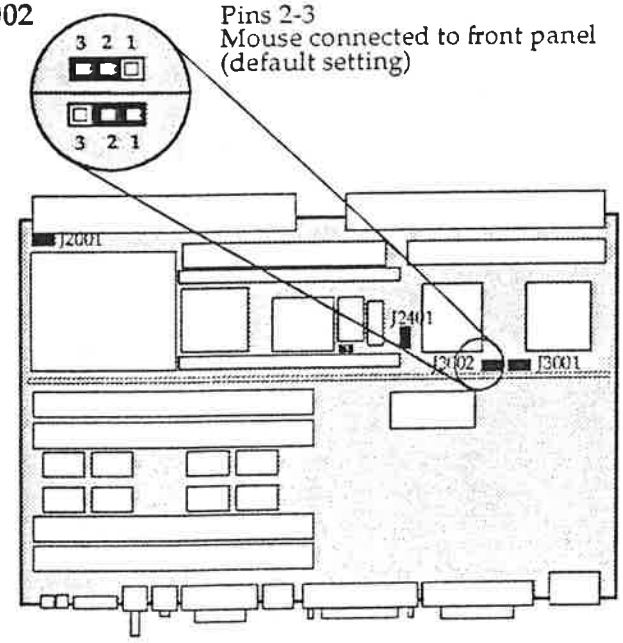
Pins 1-2
Port ~~B~~ C available on P2



Pins 2-3
Keyboard connected to front panel
(default setting)

Jumper J3002

Pins 1-2
Port ~~C~~ D available on P2

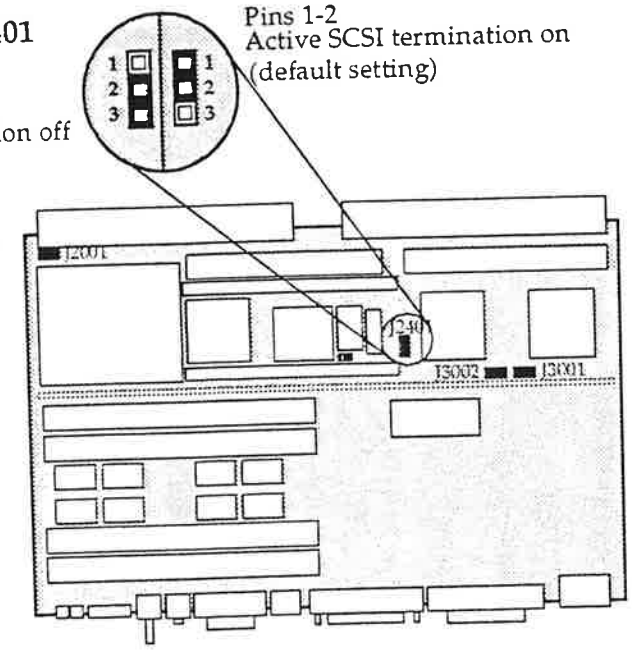


Pins 2-3
Mouse connected to front panel
(default setting)

Jumper J2401

Pins 1-2
Active SCSI termination on
(default setting)

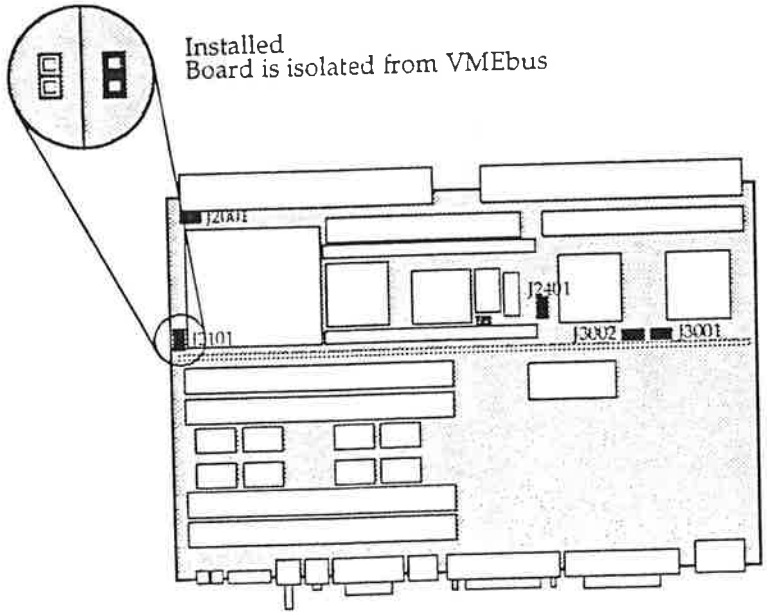
Pins 2-3
Active SCSI termination off



Jumper J2101

Removed
VME interface is active
(default setting)

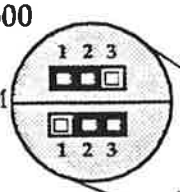
Installed
Board is isolated from VMEbus



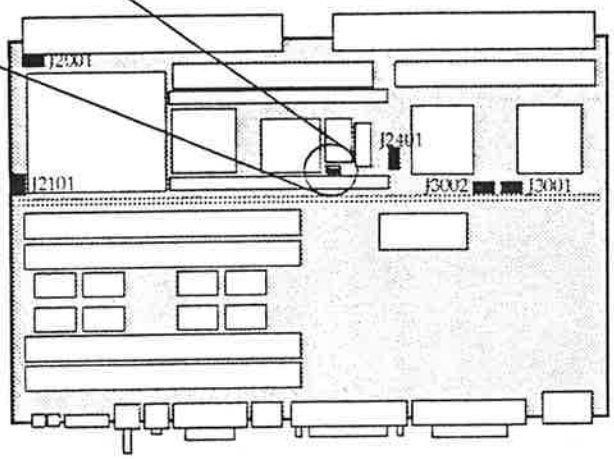
Solder Bead F4500

Beads 2-3
+5V/+12V FLASH EPROM

Beads 1-2
+5V FLASH EPROM only



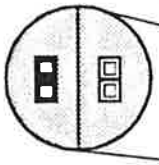
Factory setting. Shall not be changed by the user.

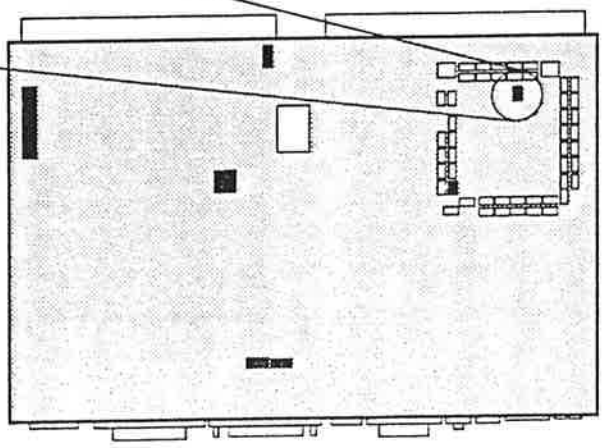


Solder Bead F5500

Open
VME BERR signal disconnected from SCV64

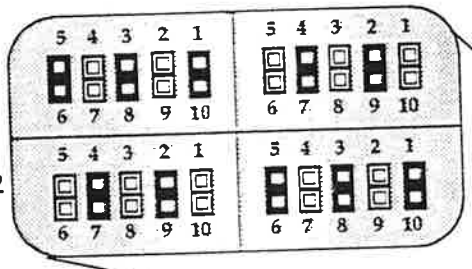
Installed
VME BERR signal connected to SCV64
(default setting)





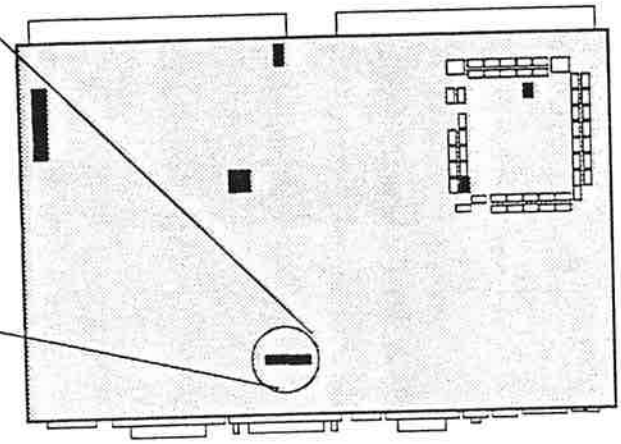
Solder Beads

IC3008B IC3008A



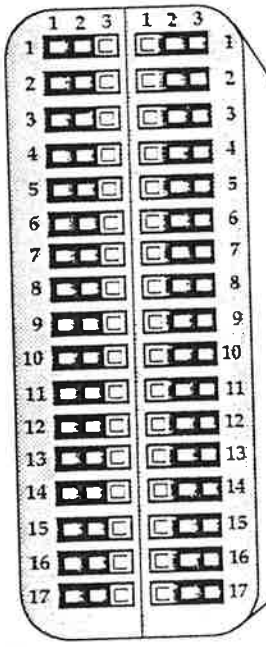
Channel B is RS232
(default setting)

Channel B is RS422

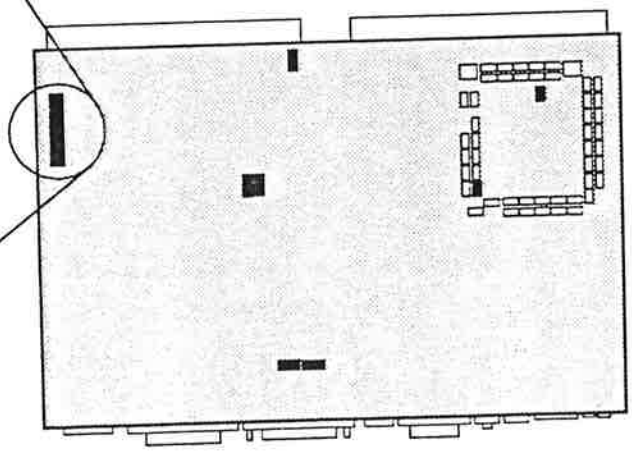


Solder Beads SB5600

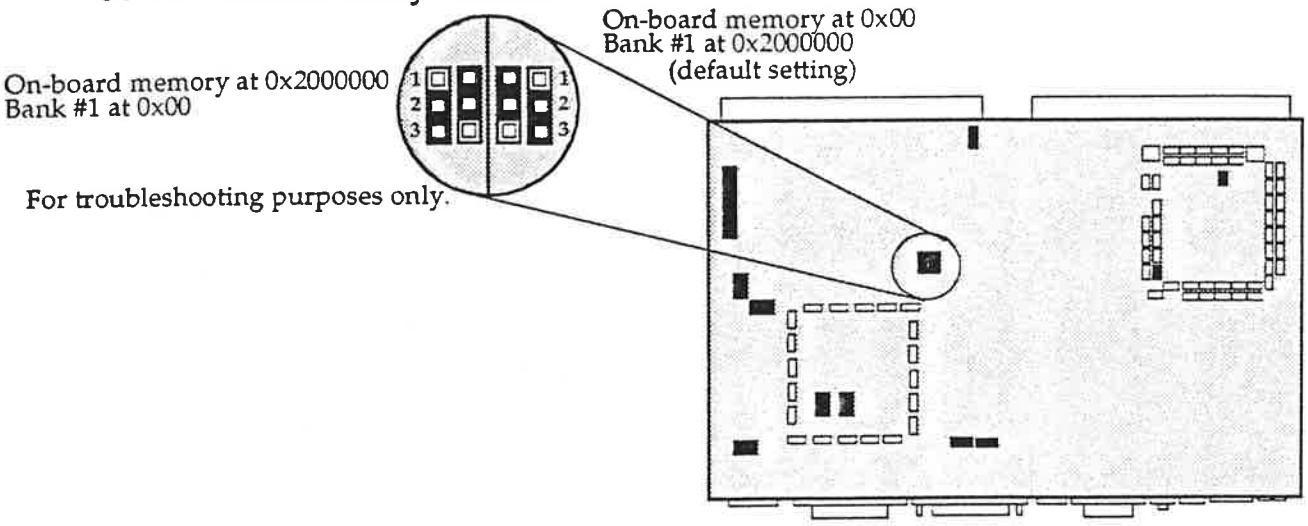
All Pins 1-2
Printer interface on P2
(default setting)



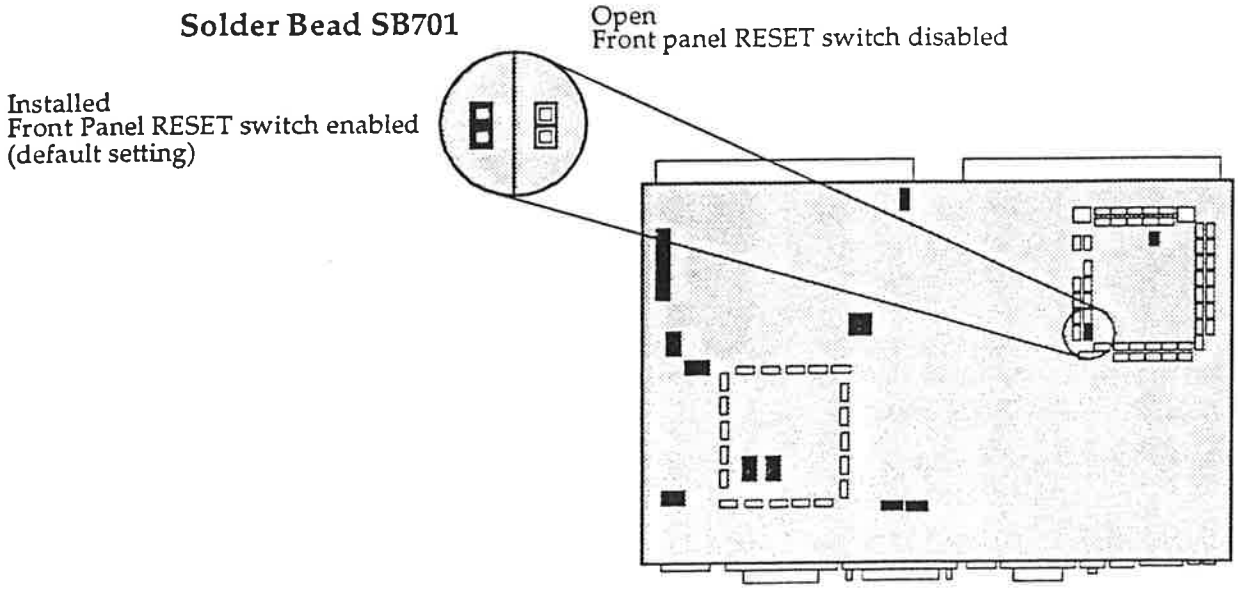
All Pins 2-3
Floppy interface on P2



Solder Beads Memory Address



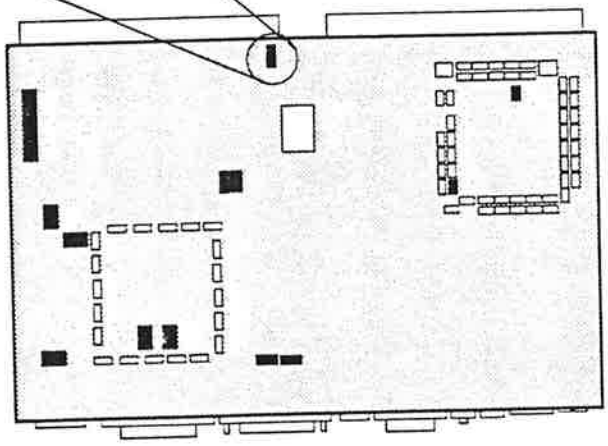
Solder Bead SB701



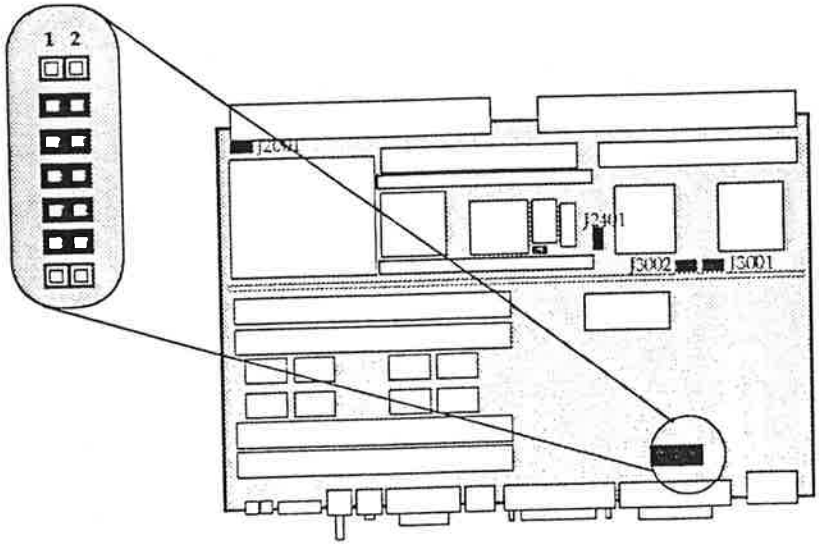
Solder Bead SB5601

Open
SCSI termination power disconnected from P2
(default setting)

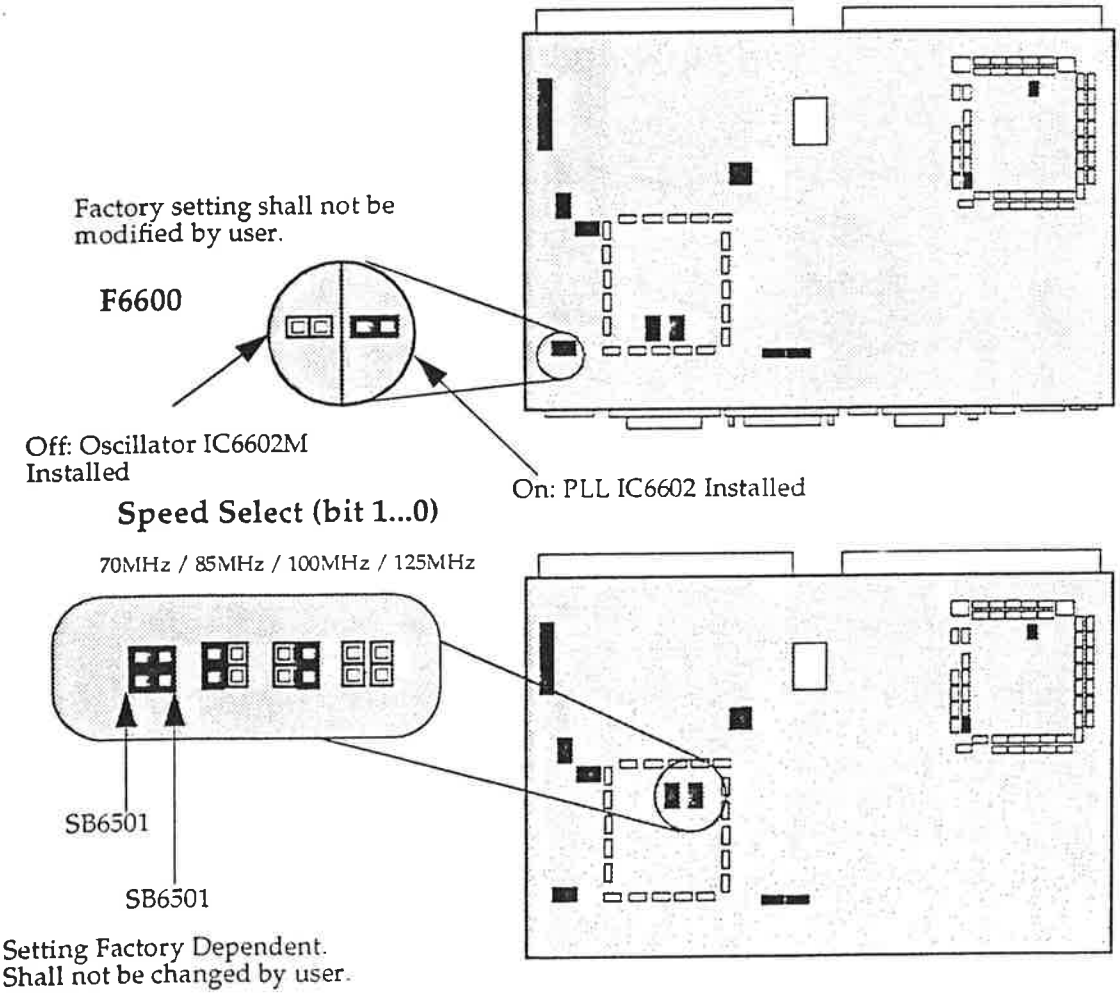
Installed
SCSI Termination power on P2



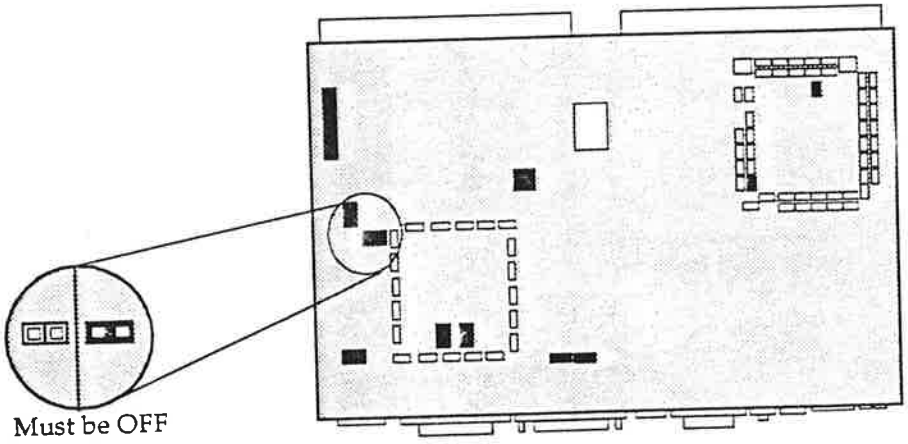
JTAG Connector



Bottom CPU Clock Select


Factory setting shall not be modified by user.
F6600
Off: Oscillator IC6602M Installed
On: PLL IC6602 Installed
Speed Select (bit 1...0)
70MHz / 85MHz / 100MHz / 125MHz
SB6501
SB6501
Setting Factory Dependent. Shall not be changed by user.

Speed Select bit 3-2



Themis SPARC 5/64 Paddle Board

The paddle board for the Themis 5/64 is an adapter board that provides industry standard connectors for accessing the interface signals available on the user pins of the P2 VMEbus connector.

The board attaches to the rear of the P2 backplane and occupies two slots. There are two types of assemblies:

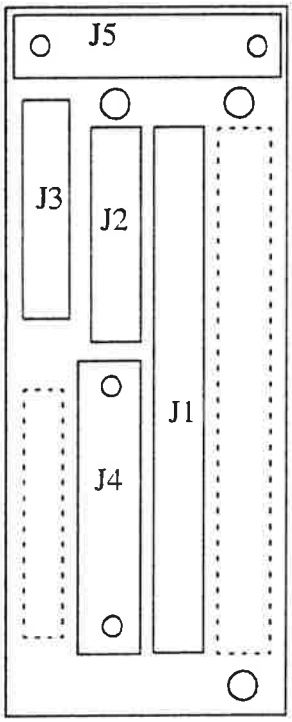
- one for AUI, Port C/D, SCSI-2, and Centronics interfaces
- the other for AUI, Port C/D, SCSI-2, and Floppy interfaces

Connectors for the following interfaces are provided on the paddle board. (The Themis part number for the 5/64 paddle board with floppy interfaces is 102311, With Centronics interface 102313)

Interface	ID	Connector Type	Manufacturers Name	Manufacturers Part Number	Themis Part #
AUI Ethernet	J5	DB-15	AMP	747299-8	101510
Centronics Parallel	J4	IBM Compatible Female DB-25	AMP	745967-7	101914
SCSI-2	J1	50-pin IDC	AMP	1-499922-0	101509
Floppy	J6	34-pin IDC	AMP		
Serial Port C	~~J2~~ J3	14-pin IDC	AMP	499922-2	101912
Serial Port D	~~J3~~ J2	14-pin IDC	AMP	499922-2	101912

J5 Ethernet Connector

Pin	Name	Dir	Description
1	GND	-	Ground
2	CP	In	Collision+
3	TP	Out	Transmit Data+
4	GND	-	Ground
5	RP	In	Receive Data+
6	GND	-	Ground
7			
8			
9	CN	In	Collision-
10	TN	Out	Transmit Data-
11	GND	-	Ground
12	RN	In	Receive Data-
13	+12V	Out	12V Ethernet Power
14		-	N/C
15		-	N/C



J4 Centronics Parallel Connector

Pin	Name	Description
1	STROBE-	
2	LPDATA1	
3	LPDADA2	
4	LPDATA3	
5	LPDADA4	
6	LPDATA5	
7	LPDATA6	
8	LPDATA7	
9	LPDADA8	
10	ACK-	
11	BUSY	
12	EMPTY	
13	SELECT	
14	AUTOFEED	
15	ERROR-	
16	INIT-	
17	SELIN-	
18	GND	Ground
19	GND	Ground
20	GND	Ground
21	GND	Ground
22	GND	Ground
23	GND	Ground
24	GND	Ground
25	GND	Ground

OpenBoot PROM Extension Commands

I.1 OBP Environment Variables

The Themis 5/64 implements a full VMEbus interface using the Newbridge SCV64 VMEbus controller implemented with the KSIC SBus to SCV64 interface. To control the VMEbus related systems settings, a number of environment variables have been added to the Sun OpenBoot PROM (OPB).

Variable	Default	Description
`vme32-slave-base`	0	Set VME slave base address in A32 mode
`vme24-slave-base`	0	Set VME slave base address in A24 mode
`cf-slot0`	`ssssssssssssssss`	Specifies alloaction of each 16MB Sbus segment to SBus (s) or VMEbus (v) for the 256MB SBus slot 0 address range.
`cf-slot1`	`ssssssssssssssss`	Specifies alloaction of each 16MB Sbus segment to SBus (s) or VMEbus (v) for the 256MB SBus slot 1 address range
`vme-boards`		This system variable should never be altered by the user. Use the *add-vme* and *remove-vme* commands to update this variable

These variables may be setup at the **ok** OBP prompt using:

```
setenv variable_name value
```

or when running Solaris with the command:

```
eeprom variable_name=value
```

A board RESET is required for the new values to take effect.

I.2 *OBP VME and Support Commands*

The advanced VMEbus interface implements a flexible VME memory mapping scheme that allows the user to asign 16MB elements of each 256MB SBus address space selectively to access SBus or VMEbus. This map is also configured using OBP commands described below.

The Forth commands available for manipulating the on-board VME MMU contents and the FLASH devices are::

Command	Stack Arguments	Description
show-vme	(--)	Display a list of all A32 boards defined
add-vme *name*	(--)	Add an A32 board accessible by the 5/64 to the VME chassis. User will be prompted for required board information (base address...)
remove-vme	(*n* --)	Remove the *n*th board from the table
dump-vme-map	(--)	Dump contents of the VME MMU map
flash-erase	(*n* --)	Erase *n*th FLASH device (*n*=1,2). Currently only the AM28F020A is supported
flash-copy	(*addr size n* --)	Copy *size* bytes starting at location *addr* to destination *device offset* (= 0) of FLASH device *n*.(see note on next page)

You can program the Flash under OBP using simple commands. For example, if you would like to program the Flash with the VxWorks bootrom file. First, you need to declare the Ethernet address of your board in the /etc/ethers file of your Sun server. Secondly, place the file you want to copy (bootrom) in flash into the /tftpboot of your Sun. Make sure that Sun is running tftpd. Then from OBP you have to type:

ok 4000 dload bootrom

ok 4000 20 40000 memmap

ok 40000 1 flash-copy

Then, (using a small scredriver) change th evalue of the rotary switch (on the board's front panel) to be different than f. On the next reset, this will make the board boot from the Flash. To go back using OBP, change the rotary switch position to f.

I.2.1 5/64 OBP Command Description

The following sections describe each 5/64 specific OBP command and its usage in detail.

show-vme (--)

This commands displays a list of all configured A32 VMEbus boards.

add-vme name (--)

This interactive command adds or modifies an entry in the VMMU mapping table. For each VMEbus board to be configured into the system, the user is prompted for the slave VME base address, the size of the address space occuppied and the applicable access mode (A--:D--). add-vme must be run for each VME board in the chassis that is accessible to the 5/64. Below is an "add-vme" session example:

ok add-vme tsvme13x

Currently,

VME boards accessible by 5/64:

Board # Name Address Mode/Type Size

Total: 0 board(s) defined

What is the board's addressing/data mode? (1.A32D32 2.A32D16 3.others):1

Enter the board # the new board will be inserted after in the list:0

Enter base address (hexdecimal):90000000

Enter size (hexdecimal):100000

Board : tsvme13x,90000000,A32D32,100000 number: 1 is added to the list

remove-vme (n --)

This command removes entries from the VME MMU table. The user will be prompted to enter the slave VME slave address, the address space size and the access mode (A32:D--) for each board to be removed. Below is a "remove-vme" session:

ok 1 remove-vme

Board #: 1 is removed from the list

dump-vme-map (--)

This commands shows a tabular display of the VME MMU map settings for each SBus slot. Below is an example of a "dump-vme-map" session:

ok dump-vme-map

Vmmu content : SLOT # 0 ok boot disk0

slice #	Address	Content	Conf	Target	Size	VME base	SBus base
0	7e0003c0	330	1,1	Sbus	D32	30000000	30000000
1	7e0007c0	331	1,1	Sbus	D32	31000000	31000000
2	7e0003c4	332	1,1	Sbus	D32	32000000	32000000
3	7e0007c4	333	1,1	Sbus	D32	33000000	33000000
4	7e0003c8	334	1,1	Sbus	D32	34000000	34000000
5	7e0007c8	335	1,1	Sbus	D32	35000000	35000000
6	7e0003cc	336	1,1	Sbus	D32	36000000	36000000
7	7e0007cc	337	1,1	Sbus	D32	37000000	37000000
8	7e0003d0	338	1,1	Sbus	D32	38000000	38000000
9	7e0007d0	339	1,1	Sbus	D32	39000000	39000000
a	7e0003d4	33a	1,1	Sbus	D32	3a000000	3a000000
b	7e0007d4	33b	1,1	Sbus	D32	3b000000	3b000000
c	7e0003d8	33c	1,1	Sbus	D32	3c000000	3c000000
d	7e0007d8	33d	1,1	Sbus	D32	3d000000	3d000000
e	7e0003dc	33e	1,1	Sbus	D32	3e000000	3e000000
f	7e0007dc	33f	1,1	Sbus	D32	3f000000	3f000000

Vmmu content : SLOT # 1

slice #	Address	Content	Conf	Target	Size	VME base	SBus base
0	7e0003a0	340	1,1	Sbus	D32	40000000	40000000
1	7e0007a0	341	1,1	Sbus	D32	41000000	41000000
2	7e0003a4	342	1,1	Sbus	D32	42000000	42000000
3	7e0007a4	343	1,1	Sbus	D32	43000000	43000000
4	7e0003a8	344	1,1	Sbus	D32	44000000	44000000
5	7e0007a8	345	1,1	Sbus	D32	45000000	45000000
6	7e0003ac	346	1,1	Sbus	D32	46000000	46000000
7	7e0007ac	347	1,1	Sbus	D32	47000000	47000000
8	7e0003b0	348	1,1	Sbus	D32	48000000	48000000
9	7e0007b0	349	1,1	Sbus	D32	49000000	49000000
a	7e0003b4	34a	1,1	Sbus	D32	4a000000	4a000000
b	7e0007b4	34b	1,1	Sbus	D32	4b000000	4b000000
c	7e0003b8	34c	1,1	Sbus	D32	4c000000	4c000000
d	7e0007b8	34d	1,1	Sbus	D32	4d000000	4d000000
e	7e0003bc	34e	1,1	Sbus	D32	4e000000	4e000000
f	7e0007bc	34f	1,1	Sbus	D32	4f000000	4f000000

Vmmu content: SLOT #2

PTH OF VMMU - DESCRIPTOR TABLE

slice #	Address	Content	Conf	Target	Size	VME base	SBus base
0	7e000360	ff	1,1	VME A32	D16	ff000000	50000000
1	7e000760	ff	1,1	VME A32	D16	ff000000	51000000
2	7e000364	ff	1,1	VME A32	D16	ff000000	52000000
3	7e000764	ff	1,1	VME A32	D16	ff000000	53000000
4	7e000368	ff	1,1	VME A32	D16	ff000000	54000000
5	7e000768	ff	1,1	VME A32	D16	ff000000	55000000
6	7e00036c	ff	1,1	VME A32	D16	ff000000	56000000
7	7e00076c	ff	1,1	VME A32	D16	ff000000	57000000
8	7e000370	ff	1,1	VME A32	D16	ff000000	58000000
9	7e000770	ff	1,1	VME A32	D16	ff000000	59000000
a	7e000374	ff	1,1	VME A32	D16	ff000000	5a000000
b	7e000774	ff	1,1	VME A32	D16	ff000000	5b000000
c	7e000378	ff	1,1	VME A32	D16	ff000000	5c000000
d	7e000778	ff	1,1	VME A32	D16	ff000000	5d000000
e	7e00037c	ff	1,1	VME A32	D16	ff000000	5e000000
f	7e00077c	ff	1,1	VME A32	D16	ff000000	5f000000

Vmmu content: SLOT #3

slice #	Address	Content	Conf	Target	Size	VME base	SBus base
0	7e0002e0	ff	1,1	VME A32	D16	ff000000	60000000
1	7e0006e0	ff	1,1	VME A32	D16	ff000000	61000000
2	7e0002e4	ff	1,1	VME A32	D16	ff000000	62000000
3	7e0006e4	ff	1,1	VME A32	D16	ff000000	63000000
4	7e0002e8	ff	1,1	VME A32	D16	ff000000	64000000
5	7e0006e8	ff	1,1	VME A32	D16	ff000000	65000000
6	7e0002ec	ff	1,1	VME A32	D16	ff000000	66000000
7	7e0006ec	ff	1,1	VME A32	D16	ff000000	67000000
8	7e0002f4	ff	1,1	VME A32	D16	ff000000	68000000
9	7e0006f0	ff	1,1	VME A32	D16	ff000000	69000000
a	7e0002f4	ff	1,1	VME A32	D16	ff000000	6a000000
b	7e0006f4	ff	1,1	VME A32	D16	ff000000	6b000000
c	7e0002f8	ff	1,1	VME A32	D16	ff000000	6c000000
d	7e0006f8	ff	1,1	VME A32	D16	ff000000	6d000000
e	7e0002fc	ff	1,1	VME A32	D16	ff000000	6e000000
f	7e0006fc	ff	1,1	VME A32	D16	ff000000	6f000000

Vmmu content: SLOT #4

slice #	Address	Content	Conf	Target	Size	VME base	SBus base
0	7e0001e0	370	1,1	Sbus SLAVIO	D32	70000000	70000000
1	7e0005e0	371	1,1	Sbus SLAVIO	D32	71000000	71000000
2	7e0001e4	ff	1,1	VME A32	D16	ff000000	72000000
3	7e0005e4	ff	1,1	VME A32	D16	ff000000	73000000
4	7e0001e8	ff	1,1	VME A32	D16	ff000000	74000000
5	7e0005e8	ff	1,1	VME A24	D32	f9000000	75000000
6	7e0001ec	f8	1,1	VME A24	D16	f8000000	76000000
7	7e0005ec	ff	1,1	VME A16	D16	ff000000	77000000
8	7e0001f0	378	1,1	Sbus MACIO	D32	78000000	78000000
9	7e0005f0	379	1,1	Sbus MACIO	D32	79000000	79000000
a	7e0001f4	37a	1,1	Sbus MACIO	D32	7a000000	7a000000
b	7e0005f4	37b	1,1	Sbus MACIO	D32	7b000000	7b000000
c	7e0001f8	37c	1,1	Sbus MACIO	D32	7c000000	7c000000
d	7e0005f8	37d	1,1	Sbus SC64	D32	7d000000	7d000000
e	7e0001fc	37e	1,1	Sbus VMEMMU	D32	7e000000	7e000000
f	7e0005fc	37f	1,1	Sbus IACK	D32	7f000000	7f000000

I.2.2 5/64 OBP Environment Variables Description

The following sections describe each 5/64 specific OBP environment variables, their settings and usage in detail.

vme32-slave-base

This variables determines the slave base address of the 5/64 in A32 addressing mode. The 5/64 DVMA address space will be accessible to other VMEbus masters at the specified address. Specify the *vme32-slave-base* value using the *setenv* command.

```
ok> setenv vme32-slave-base 1000.0000
ok>
```

vme24-slave-base

This variables determines the slave base address of the 5/64 in A24 addressing mode. The 5/64 DVMA address space will be accessible to other VMEbus masters at the specified address. Specify the *vme24-slave-base* value using the *setenv* command.

```
ok> setenv vme24-slave-base d0.0000
ok>
```

cf-slot0

To facility a flexible VMEbus addressing scheme, each 256MB address space of all SBus slots are segmented into 16 sections of 16MB each. The cf-slot variable allows the user to conveniently assign each of the 16 segments to the Sbus or VMEbus by specifying a 16 character text string composed of 's' and 'v' characters indication Sbus or VMEbus.

The following command configures SBus slot #0 for VMEbus access only (this assumes no SBus card is installed):

If an installed SBus card occupies only part the the available SBus address space, the unused segments can be allocated to the VMEbus as indicated in the following example, which allocates the first 56MB to Sbus and the remaining 200MB to the VMEbus:

```
ok> setenv cf-slot0 vvvvvvvvvvvvvvv
ok>
```

```
ok> setenv cf-slot0 ssssvvvvvvvvvvv
ok>
```

cf-slot1

SBUs slot #1 is configured similar to Sbus slot #0 described above.

Index

A

Address Mapping Specifications 109
Addressability 61
add-vme name 143
Arbitration Modes 92
Attaching a Keyboard and Mouse 53
Attaching the 5/64 To A Network 52
AUI Ethernet 25
AUI Ethernet Connector 108

B

Bus Arbitration 92
Bus Timer 93

C

cf-slot0 150
cf-slot1 151
Components from other vendors 9
Configuring the SPARC 5/64 35
Configuring The VME Interface 41
Connectors 9
Counter/Timer Controls 20
Customer Support 33

D

DMA Controller Macrocell 17
DMA Transfers 98
DMALAR DMA Local Address Register 98
DMATC DMA Transfer Count Register 99
DMAVAR DMA VMEbus Address Register 98
DMAVTC DMA VMEbus Transfer Count Register 99
DRAM Addressing Window 113
dump-vme-map 145

E

EBus Controller 19
Ethernet Controller Macrocell 17

F

Factory Default Settings 123
Features 88
FLASH EPROM 24
FLASH Programming Voltage Control Register 116
FLASH Select Register 117
Floppy Disk Controller 18

G

General Description 90
grounding strap, attaching 34

H

How To Start Quickly 5
How to Use This Manual 1

I

In Case Of Difficulties 5
Input/Output 28
Input/Output Controllers and 8-Bit Devices 17
Installing An SBus Card 47
Installing SIMM Memory 49
Installing The 5/64 Paddle Board 47
Integer Unit (IU) and Floating-Point Unit (FPU) 14
Intended Audience 2
Interrupt Controls 19
Introduction to the Themis SPARC 5/64 7
IVECT Register 96

J

JTAG Internal & Boundary Scan Controller 19

K

Keyboard/Mouse
    8-Pin DIN Connector (P3002) 106
Keyboard/Mouse Controller 18
KSIC Bottom 80
KSIC SBus to SCV64 Interface 20
KSIC Top 79

KSIC/SCV64 Interrupt Status Register 118

L

LMFIFO Location Monitor Read Port 98
Local SCV64 Interrupt Acknowledge Registers 120
Location Monitor 97

M

MACIO (Master I/O Controller) 17
MACIO Mapping 115
Main Memory Cache 14
Main Memory DRAM SIMM Female Connectors 24
Master VME Address Generation 109
Master VME MMU and SBus Slot Address Mapping 110
Microprocessor Performance Figures (85MHz) 27
microSPARC II Mapping 112
microSPARC II Microprocessor 14
Mouse/Keyboard Connector 25

N

Non-Operating Environmental Requirements For This Product 32
Non-volatile RAM (NVRAM) 23

O

OBP Environment Variables 141
OBP VME and Support Commands 142
On-board Main Memory 29
Open Boot PROM (OBP) 23
OpenBOOT PROM Configuration 38
OpenBoot PROM Extension Commands 141
Operating Environmental Requirements For This Product 32
Organization 3

P

Performance 8
Physical Dimensions 27
Power Requirements For This Product 31
Primary Features 56
Printer Port Controller Macrocell 17

R

RAM Controller 15
Reference Memory Management Unit (MMU) 15
Registering Your SPARC 5/64 33

remove-vme 144
Required Initial Power-Up System Components 25
RISC Microprocessor 27
Rotary Switch Register 116

S

SBus 30
SBus Address Mapping 44
SBus Connectors Pinout
J602 (Slot 1) and J601(Slot 0) 105
SBus Controller 15
SBus Slot #0 44
SBus Slot #1 44
SBus Slot #2 44
SBus Slot #3 44
SBus slot #4 45
SBus Slot (SBus Devices) Connectors 24
SCSI Connector
High-Density 50-pin (P2401) 107
SCSI Female Connector 24
SCSI-2 Controller Macrocell 17
SCV64 Inter-Processor Communication/Location Monitor 72
SCV64 Registers 91
SCV64 VMEbus Controller 20
Serial Port A Controller 18
Serial Port A&B 25
Serial Port B Controller 18
Serial Ports A and B
26-pin Connector (P3001) 106
show-vme 143
Single Inline Memory Modules (DRAM SIMMs) 21
SLAVIO (Slave I/O Controller) 18
SLAVIO/EBus Mapping 114
Slot 0
Extension SBus Slot 20
Slot 1
Extension SBus Slot 21
SPARC 5/64 Bottom 82
SPARC 5/64 Top 81
System Considerations 25
System Controller 57
System Reset Controls 20

T

Themis SPARC 5/64 Board 11, 27
Themis SPARC 5/64 Board Drawings 79
Themis SPARC 5/64 Description 11
Themis SPARC 5/64 Electrical Specifications 31
Themis SPARC 5/64 Environmental Specifications 31
Themis SPARC 5/64 Features 8
Themis SPARC 5/64 Front Panels 83
Themis SPARC 5/64 Installation 33
Themis SPARC 5/64 Main Components 14
Themis SPARC 5/64 Programmers Guide 55
Themis SPARC 5/64 Specifications 27
Themis SPARC 5/64 Supported Standards 10
Themis SPARC 5/64-2S2E Paddle Board 135
Time-of-Day Clock (TOD) 24
Translation Lookaside Buffer (TLB) 15

U

Unpacking The SPARC 5/64 34
User LED Register 116

V

VINT Register 95
VME Access Speed 85
VME Bus Connectors (P1) 102
VME Bus Floppy Interface Connectors Continued (P2) 104
VME Bus Printer Interface Connectors(P2) 103
VME Interrupt Acknowledge Registers 119
VME Interrupt Handling 96
VME Interrupt Status 71
VME Interrupter 95
VME MMU Address Translation Table 45, 61
VME Slave Interface 94
VME System Controller 92
vme24-slave-base 150
vme32-slave-base 150
VMEBAR Register 94
VMEbus 30
VMEBus Interface Implementation 87
VMEbus Interrupt Acknowledgment 71
VMEbus Master Interface 58
VMEbus Memory Allocation 43
VMEbus Slave Interface 65
VMEbus/SCV64 Interrupts 69
VMMU Enable Register 121

W

wrist strap, attaching 34